

Jarno Lehtorinne

Medicine Time -lääkemuistutin

Sovelluksen kehitys Windows Phone 7.5 alustalle

Medicine Time -lääkemuistutin

Sovelluksen kehitys Windows Phone 7.5 alustalle

Jarno Lehtorinne
Opinnäytetyö
Kevät 2012
Tietotekniikka
Oulun seudun ammattikorkeakoulu

ALKULAUSE

Tämä opinnäytetyö tehtiin Leo Ilkon toimeksiannosta kevään 2012 aikana Oulun seudun ammattikorkeakoulun Raahen kampuksella. Työ on osa ubiHOME-projektia. Tehtäväni oli tehdä ubiPILL-järjestelmän hälytinosan erillISRatkaisu, joka voidaan jatkokehityksenä myöhemmin liittää ubiPILL -kokonaisuuteen. Tehtävänäni oli myös laittaa sovellus myyntiin Windows Phone Marketplace -palveluun.

Opinnäytetyöni pohjautuu osittain Ari Tandoganin opinnäytetyöhön ubiPILL-lääkeannostelijan uuden version suunnitelmasta.

Haluan kiittää työtä valvonutta ohjaajaa Leo Ilkkoa tuesta ja ohjauksesta.

Kajaanissa 31.5.2012

Jarno Lehtorinne

TIIVISTELMÄ

Oulun seudun ammattikorkeakoulu
Tietotekniikan koulutusohjelma, Mobiiliteknologiat

Tekijä: Jarno Lehtorinne

Opinnäytetyön nimi: Medicine Time -lääkemuistutin

Työn ohjaaja: Leo Ilkko

Työn valmistumislukukausi ja -vuosi: Kevät 2012

Sivumäärä:57

Oulun seudun ammattikorkeakoulun ubiHOME-laboratorio kehittää yksinelävien vanhusten tarpeisiin soveltuvaa älytekniikkaa, ja ubiPILL-lääkeannostelija on yksi näistä projekteista. Opinnäytetyön tavoitteena oli kehittää erillinen sovellus Windows Phone 7.5 -alustalle, joka pohjautuu samaan ideaan kuin ubiPILL.

UbiPILL-järjestelmän aikaisemmat versiot koostuivat päätelaitteesta, henkilökohtaisesta hälyttimestä ja palvelimesta. Medicine Time -sovellus ei tarvitse erillistä päätelaitetta, vaan hälyttää vanhukselle, milloin heidän pitää ottaa lääkkeensä ja merkitsee sovelluksen sisäiseen lokiin tiedon, onko lääkkeet otettu ja mihin aikaan.

Työn toteutusta varten tutustuin aikaisempiin opinnäytetöihin, jotka käsittelivät ubiPILL-järjestelmää, sekä tutustuin Windows Phone -alustan ohjelmointiin internetistä löytyvien dokumenttien ja harjoitusten kautta. Sovelluksen kehittämisen lisäksi työhöni kuului Windows Phone -alustan sovelluksen elinkaaren esittely ohjelmoinnin aloittamisesta aina Marketplace-palveluun myyntiin asettamiseen saakka.

Opinnäytetyön tuloksena Medicine Time -sovellus valmistui, se asetettiin myyntiin Marketplaceen, ja sen lähdekoodi jää koululle mahdollista jatkokehitystä varten. Jatkokehitysmahdollisuuksia olisivat esimerkiksi ohjelmoida Medicine Time toimimaan erillisen serverisovelluksen kanssa, jonka kautta omaiset voivat tarkastella lääkkeenottotietoja Internetissä, sekä Bluetooth-yhteys dosetissa olevaan päätelaitteeseen.

Asiasanat:

ubiPILL, Medicine Time, Lääkemuistutin, Windows Phone

ABSTRACT

Oulu University of Applied Sciences
Degree Programme in Information Technology, Mobile Technologies

Author: Jarno Lehtorinne

Title of thesis: Medicine Time -Medicine Reminder

Supervisor: Leo Ilkko

Term and year of completion: Spring 2012

Number of pages:57

Oulu University of Applied Sciences UbiHOME-laboratory is developing smart technology for needs of independently living elders. UbiPILL-Medicine dispenser is one of these projects.

The aim of this thesis was to create a separate Windows Phone Application that is based on the same idea as ubiPILL.

The earlier versions of ubiPILL consisted of a terminal, a personal alarmer and a server. Medicine Time -application does not need a separate terminal, but it alerts elders when they have to take their medicine and writes to the inner log of the application that the medicines been taken and at what time.

The work was started by reading previous theses about ubiPILL and familiarizing myself with a Windows Phone programming with documents and practices found in the internet. One part of my work was also to present the lifecycle of a Windows Phone application from start of the programming to uploading the application to Marketplace.

As a result of this thesis, the Medicine Time -application was completed and uploaded to Marketplace, and its source code will remain at school for a possible further development.

Possibilities of further development would be for example to program Medicine Time to work with a separate server application, from where relatives could view the dosing information in the internet, and a Bluetooth-connection to the internal terminal of dosette.

Keywords:

ubiPILL, Medicine Time, Medicine Reminder, Windows Phone

Sisällys

1 JOHDANTO	7
1.1 Työn rakenne	7
1.2 ubiHOME	8
1.3 ubiPILL-lääkeannostelija	8
2 MÄÄRITELMÄ	10
2.1 Sovelluksen määritelmä	10
2.1.1 Hälyttimen käyttöliittymän määritelmä	10
2.1.2 Asetusten käyttöliittymän määritelmä	11
2.1.3 Lokin käyttöliittymän määritelmä	11
2.2 Tutustuminen Windows Phone -alustaan	12
2.3 Ohjelmistosuunnittelu	12
3 TOIMINTAYMPÄRISTÖ	15
3.1 Windows Phone	15
3.1.2 Windows Phone Marketplace	15
3.1.3 .NET	15
3.1.4 Microsoft Silverlight	16
3.1.5 XAML	16
3.2 Windows Live SkyDrive	16
3.3 Microsoft Visual Studio 2010	17
3.4 Nokia Lumia	18
3.5 Microsoft Expression Blend 4	19
4 SOVELLUKSEN TOTEUTUS	21
4.1 Windows Phone -sovelluksen kehittäminen Visual Studiolla	21
4.1.1 Käyttöliittymän suunnittelu	23
4.1.2 Toiminnallisuuden suunnittelu	25
4.1.3 Käyttöliittymän viimeistely	26
4.2 Medicine Time -sovelluksen toteutus	30
4.2.1 Hälyttimen toiminnallisuuden suunnittelu	31
4.2.2 Lokikirjan toiminnallisuuden suunnittelu	33
4.2.3 Asetusten toiminnallisuuden suunnittelu	35
4.2.4 Muistutusten toiminnallisuuden suunnittelu	36

4.2.5 Käyttöliittymän suunnittelu	41
4.3 Muutoksia ja ongelmia	41
4.3.1 Tekstiviestitoiminnon poisjättäminen	41
4.3.2 SkyDrive-toiminnon tutkiminen	42
4.3.3 Reminder-toiminnon ongelmat	44
5 SOVELLUKSEN TESTAUS JA JULKAISU	46
5.1 Sovelluksen testaus emulaattorilla	46
5.2 Sovelluksen testaus testikäyttäjällä	46
5.3 Testausversion arkkitehtuuri	47
5.4 Lopullisen version arkkitehtuuri	50
5.5 Sovelluksen julkaisu Windows Phone Marketplace -palvelussa	54
6 JATKOKEHITYSMAHDOLLISUUDET	55
7 YHTEENVETO	56
LÄHDELUETTELO	57

1 JOHDANTO

Yksi suurimmista matkapuhelinvalmistajista Nokia otti osana strategia-muutostaan käyttöön pääasiallisena älypuhelin-alustana Windows Phonen. Tämä herätti kiinnostukseni kyseiseen alustaan, joka ei ole ollut markkinoilla kuin vasta vähän aikaa, joten kun minulle tarjottiin opinnäytetyön aiheeksi ohjelmoida sovellus tälle alustalle, otin työn kiinnostuneena vastaan.

Opinnäytetyöni aiheena oli suunnitella matkapuhelinsovellus Windows Phone -alustan versiolle 7.5. Sovellus olisi ubiPILL-lääkeannostelijan hälytinosan erillISRatkaisu, joka myöhemmin voitaisiin mahdollisesti liittää ubiPILL-kokonaisuuteen.

Osana opinnäytetyötäni oli sovelluksen kehittämisen lisäksi kertominen sovelluksen kehittämisen eri vaiheista Windows Phone -alustalle ohjelmoinnin aloittamisesta aina Windows Phone Marketplace -palveluun asettamiseen saakka. En ollut aikaisemmin kehittänyt kaupallisia sovelluksia Windows Phone -alustalle, joten tämän tutkiminen oli erityisen kiinnostava osa opinnäytetyötäni.

1.1 Työn rakenne

Työn toisessa luvussa määritellään Medicine Time -sovelluksen vaatimukset sekä ohjelmistosuunnittelumallia.

Työn kolmannessa luvussa esitellään, millaiseen toimintaympäristöön sovellus kehitetään ja millaisia työkaluja käyttämällä sovellus on kehitetty.

Työn neljännessä luvussa esitellään Windows Phone -alustan sovelluskehitystä ja kerrotaan, kuinka Medicine Time on toteutettu.

Viidennessä luvussa esitellään sovelluksen testausta ja kerrotaan sovelluksen julkaisemisesta Marketplace-palvelussa.

Kuudennessa luvussa pohditaan sovellukselle mahdollisia jatkokehitysmahdollisuuksia.

1.2 ubiHOME

UbiHOME-laboratorio sijaitsee Oulun seudun ammattikorkeakoulun Raahen kampuksella, ja siellä kehitellään älytekniikkaa helpottamaan yksinasuvien vanhusten elämää.

1.3 ubiPILL-lääkeannostelija

UbiPILL-lääkeannostelija on osa ubiHOME-järjestelmää. Lääkeannostelijan tehtävänä on muistuttaa lääkkeiden ottoajoista sekä kirjata palvelimelle tietoja siitä, onko käyttäjä ottanut lääkkeitä ajallaan vai onko lääkkeitä jäänyt ottamatta. UbiPILL-lääkeannostelijan esittely kuvassa 1.



ubiPILL – Intelligent Pill Box of Ubiquitous Home Environment

The Intelligent Pill Box project aims at developing a commercial, intelligent medicine reminder. The product helps people to take their medicine on time, reminding, keeping track and notifying for required actions. It can be linked with variety of stakeholder networks to allow members of Authorized Group of Interest – consisting e.g. of healthcare professionals, relatives and others – to be aware of the patient's health in real-time.

Authorized Group of Interest (AGI)
User interfaces supported by a wide
range of devices for setting up
alarms and viewing pill-taking activity



Mobile/Tablet Application



Web Application



ubiPILLSERVER –
Provider of Web Application and
data on pill-taking activity for
Mobile/Tablet/Web Applications

3G/4G Mobile Broadband
WLAN-to-Broadband



Table Unit –
Intelligent Home Device for alarm
indication, optionally featured with
inductive charging of battery

4th Gen Bluetooth



Alarm Unit –
Wearable device of patient's choice
for alarm indication, optionally
featured with e.g. a positioning
system or heart-rate monitor

OULU UNIVERSITY OF
APPLIED SCIENCES



2 MÄÄRITELMÄ

Määrittysten mukaisesti sovelluksen tärkein kohderyhmä ovat henkilöt, jotka joutuvat ottamaan päivittäin lääkkeitä ja haluavat muistutuksen, jotta lääkkeet tulevat otettua. Esimerkkityyppisessä tilanteessa sovelluksella on kaksi käyttäjää eli omainen, joka asettaa hälytykset ja valvoo lääkkeiden ottoa sovelluksesta löytyvän lokikirjan kautta, sekä loppukäyttäjä, joka saa sovellukselta muistutuksen lääkkeenotosta ja kuittaa lääkkeen otetuksi.

Sovelluksesta tulisi saada mahdollisimman yksinkertainen loppukäyttäjälle säilyttämällä silti monipuolisesti säädettävät asetukset omaiselle tai hoitohenkilökunnalle.

2.1 Sovelluksen määritelmä

Alkuperäisen määritelmän mukaan sovelluksen tulisi olla "ubiPILL-järjestelmän hälytinosan erillISRatkaisu, joka on liitettävissä myöhemmin ubiPILL-kokonaisuuteen. Sovellus on älypuhelimien itsenäinen taustasovellus, joka muistuttaa hälyttämällä lääkkeiden otosta."

2.1.1 Hälyttimen käyttöliittymän määritelmä

Alkuperäisen määritelmän mukaan hälyttimen käyttöliittymä koostuisi kahdesta painikkeesta :

A) OTA LÄÄKE

B) OTAN MYÖHEMMIN

Hälytyksen aktivoituessa tulee hälytysnäyttö taustalta esiin, A-painike vilkkuu punaisena, laite värisee ja hälyttää äänellä. Käyttäjä voi kuitata lääkkeen otetuksi painamalla A-painiketta ja samalla käyttöliittymä ilmoittaa onnistuneesta lääkkeenotosta. Tämän jälkeen sovellus palaa takaisin taustalle.

Sovellus tarkistaa lääkkeenottoajan matkapuhelimen kellosta ja tallentaa ajan lokiin, jota omaiset ja hoitohenkilökunta voivat tarkastella tarvittaessa.

Jos käyttäjä on hälytyksen aktivoituessa poissa kotoa tai kaukana dosetista, voi hän painamalla B-painiketta siirtää hälytystä. Sovellus menee näin ollen takaisin taustalle, ja hälytys aktivoituu uudelleen asetukset-sivulla määritetyn ajan kuluttua. Jos "Ota Lääke" -kuittauksia jää väliin 2 kertaa, lähettää sovellus asetukset-sivulla annettuun puhelinnumeroon tekstiviestin, jossa kerrotaan lääkkeiden oton jääneen väliin.

Jatkokehityksessä sovellus voidaan liittää osaksi ubiPILL-järjestelmää. Näin sovelluksen keräämät lokitiedot voidaan lähettää ubiPILL-serverille. Lisäksi sovellukseen voidaan lisätä bluetooh-yhteys, jonka kautta voisi kommunikoida ubiPILL-järjestelmän kanssa, jolloin lääkkeenotot kuittautuisivat automaattisesti. Tällä hetkellä sovellus ei kuittaa lääkkeenottoa automaattisesti dosetista ottamisen yhteydessä, vaan vastuu on käyttäjällä.

2.1.2 Asetusten käyttöliittymän määritelmä

Salasanan antamisen jälkeen omainen voi määritellä torkkuajastuksen keston minuuteissa sekä aamu-, päivä-, ilta- ja yölääkkeiden ottoaikavälit. Lisäksi käyttäjä voi määrittää puhelinnumeron, johon tarvittaessa sovellus voi lähettää tekstiviestin.

2.1.3 Lokin käyttöliittymän määritelmä

Käyttäjä voi itsekin halutessaan selata toteutuneita lääkkeenottoaikoja lokiruudussa. Omainen voi myös tyhjentää lokitietoja asetusten kautta.

2.2 Tutustuminen Windows Phone -alustaan

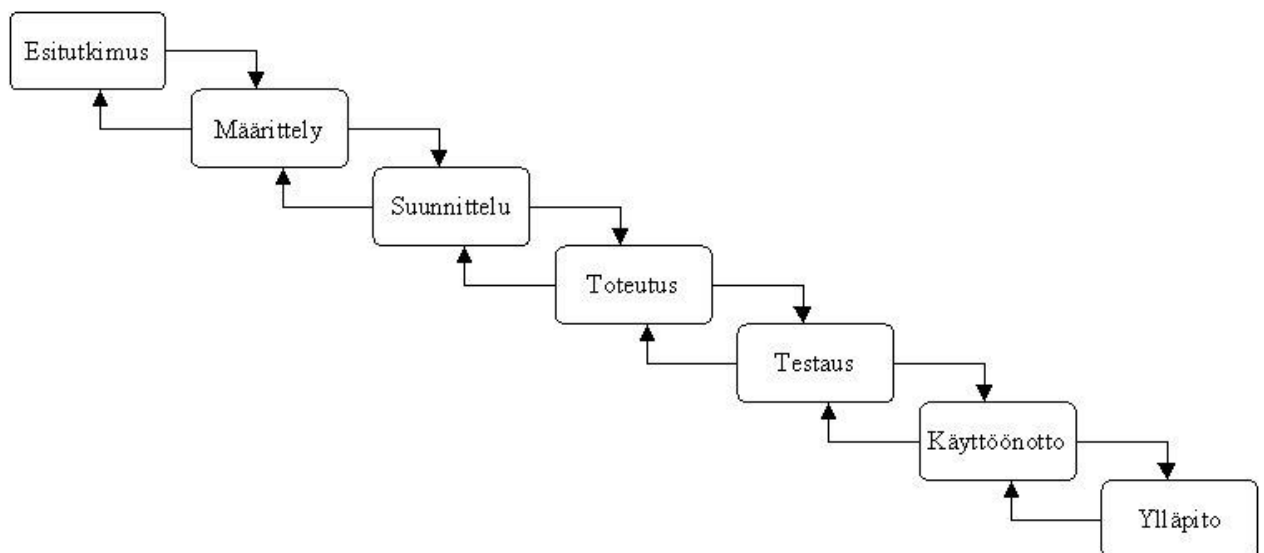
Aloitin työn tutustumalla perusteellisesti Windows Phone -alustaan. Kävin syksyllä Oulussa kuuntelemassa Microsoftin järjestämässä Geeks on Wheels -tapahtumassa eri luennoitsijoiden pitämiä esityksiä liittyen Windows Phone -alustaan. Tapahtumasta sain paljon hyödyllisiä vinkkejä sovelluskehittämisen

eri vaiheista. Lisäksi kävin syksyllä mobiilijärjestelmien ohjelmointi -kurssin, jossa Windows Phone -alustasta opetettiin perusteet.

Tämän jälkeen aloin tutustua alustaan lukemalla Microsoftin ilmaiseksi jakamaa e-kirjaa Programming Windows Phone 7 (Petzold,2010), sekä perehtymällä erilaisiin Microsoftin MSDN-sivustolla tarjoamiin harjoituksiin (Code Samples for Windows Phone 2012 , hakupäivä 4.1.2012 .)

2.3 Ohjelmistosuunnittelu

Saatuani sovelluksen perusmäärittelyt aloin kehittämään sovellusta vapaasti mukaillen vesiputousmallin vaiheita (esitelty kuvassa 2) .



KUVA 2: Esimerkki vesiputousmallista (Johdatus tietojärjestelmiin, hakupäivä 13.3.2012)

Vesiputousmallissa ohjelmistotuotannon elinkaari on jaettu vaiheenjakomalliin. "Vaiheenjakomallilla tarkoitetaan tapaa, jolla ohjelmiston kehitystyö tai koko elinkaari jaetaan vaiheisiin."(Haikala & Märijärvi 2004, 36.)

Koska kyseessä on yksittäinen projekti, lasketaan esitutkimusvaihe osaksi määrittelyvaihetta. Määrittelyvaiheessa asiakkaan toiveet analysoidaan ja niistä

johdetaan sovelluksen määrittävät vaatimukset (Haikala & Märijärvi 2004, 37-38.)

Suunnitteluvaiheessa suunnitellaan määrittelyvaiheessa annettujen vaatimusten toteutus. Suunnittelun alkuvaiheessa järjestelmä jaetaan itsenäisiksi moduuleiksi. Seuraavaksi suunnitellaan moduulien sisäinen rakenne. "Määrittelyvaiheessa kuvataan, "mitä" järjestelmä tekee, ja suunnitteluvaiheessa vastaavasti vastataan kysymykseen, "miten" se tehtävänsä suorittaa (Haikala & Märijärvi 2004, 38.)

Koska kyse on ohjelmistoprojektista, toteutusvaihetta kutsutaan ohjelmointivaiheeksi. Ohjelmointivaiheessa sovellus kirjoitetaan ensimmäiseen virheettömään käännökseen asti aikaisemmista vaiheista syntyneitä suunnitelmia käyttäen (Haikala & Märijärvi 2004, 38.)

Testausvaiheen tarkoitus on löytää ohjelmistosta virheitä. Testausta tapahtuu ohjelmointivaiheen kanssa yhtäaikaaisesti; kun testatessa löytyy virhe, korjataan se ohjelmointivaiheessa (Haikala & Märijärvi 2004, 39.)

Käyttöönottovaiheessa asiakas ottaa tuotteen käyttöön ja, jos huomaa tuotteessa puutteita tai virheitä, siirtyy tuote takaisin testausvaiheeseen. Kun tuote on kaikkien osapuolten mielestä valmis, alkaa ylläpitovaihe (Haikala & Märijärvi 2004, 39.)

Ylläpitovaiheessa voidaan sopia, hoitaako sovelluksen kehittänyt osapuoli itse ylläpidon vai hoitaako asiakas itse ylläpidon. Ylläpito on viimeinen vaihe vesiputousmallia (Haikala & Märijärvi 2004, 39.)

Käytin työssäni mukautettua versiota vesiputousmallista, jossa jaoin sovelluksen eri osioihin ja tein jokaista osiota ajaen vesiputousmallin elinkaarta edestakaisin. Varsinkin määrittely-, suunnittelu-, toteutus- ja testausvaiheet toistuivat moneen kertaan jokaista osiota tehdessä.

3 TOIMINTAYMPÄRISTÖ

3.1 Windows Phone

Windows Phone on Microsoftin matkapuhelinkäyttöjärjestelmä, joka julkistettiin 2009. Windows Phone julkaistiin syksyllä 2010 ja korvasi Windows Mobile -käyttöjärjestelmän. Microsoft teki Windows Phonen tarkoituksenaan ratkaista asiakkaidensa tarpeet yhdellä alustalla (Wikipedia 2012, hakupäivä 4.5.2012.)

3.1.1 Windows Phone Mango

Windows Phone 7.5, koodinimeltään "Mango" on uusin versio Windows Phone -käyttöjärjestelmästä. Se julkaistiin 2011 vuoden viimeisellä vuosineljänneksellä ja toi mukanaan useita parannuksia verrattuna 7.0-versioon (Wikipedia 2012, Hakupäivä 4.5.2012.) Medicine Time -sovellus tehtiin tälle versiolle Windows Phone -alustaa.

3.1.2 Windows Phone Marketplace

Windows Phone marketplace on Microsoftin luoma sovellusten myyntipaikka Windows Phone -käyttöjärjestelmää käyttäville puhelimille. Huhtikuussa 2012 erilaisia sovelluksia oli jo yli 80 000 (Windows Phone Marketplace passes 80,000 apps 2012, hakupäivä 6.4.2012.) Valmis Medicine Time -sovellus laitettiin myyntiin Windows Phone Marketplace palveluun, josta halukkaat voivat ostaa sovelluksen itselleen.

3.1.3 .NET

.NET on Microsoftin kehittämä ohjelmistokehys joka sisältää laajat koodikirjastot sekä mahdollisuuden toimia useiden eri ohjelmointikielien kanssa. Uusin versio .NET ohjelmistokehyksestä on 4.0, johon pohjautuvia sovelluksia voidaan kehittää Microsoft Visual Studio 2010 -sovelluksella (Wikipedia 2012, hakupäivä 5.5.2012).

Ohjelmistokehitys .NET-ohjelmistokehyksellä Windows Phonelle tapahtuu käyttämällä joko Visual C#- tai Visual Basic -ohjelmointikieliä Microsoft Silverlight -sovelluskehityksen kanssa.

3.1.4 Microsoft Silverlight

Microsoft Silverlight on sovelluskehitys jolla voidaan luoda verkkosovelluksia erinäisten medioiden toistoon. Silverlight on myös Windows Phonen sovelluskehitysalusta ja tätä alustaa käyttäen Medicine Time -sovellus on myös luotu. Windows Phonen Silverlight perustuu Silverlightin 4.0-versioon ja sen ohjelmoimiseen voidaan käyttää .NET-ohjelmointikehystä.

(Wikipedia 2012, hakupäivä 5.5.2012).

3.1.5 XAML

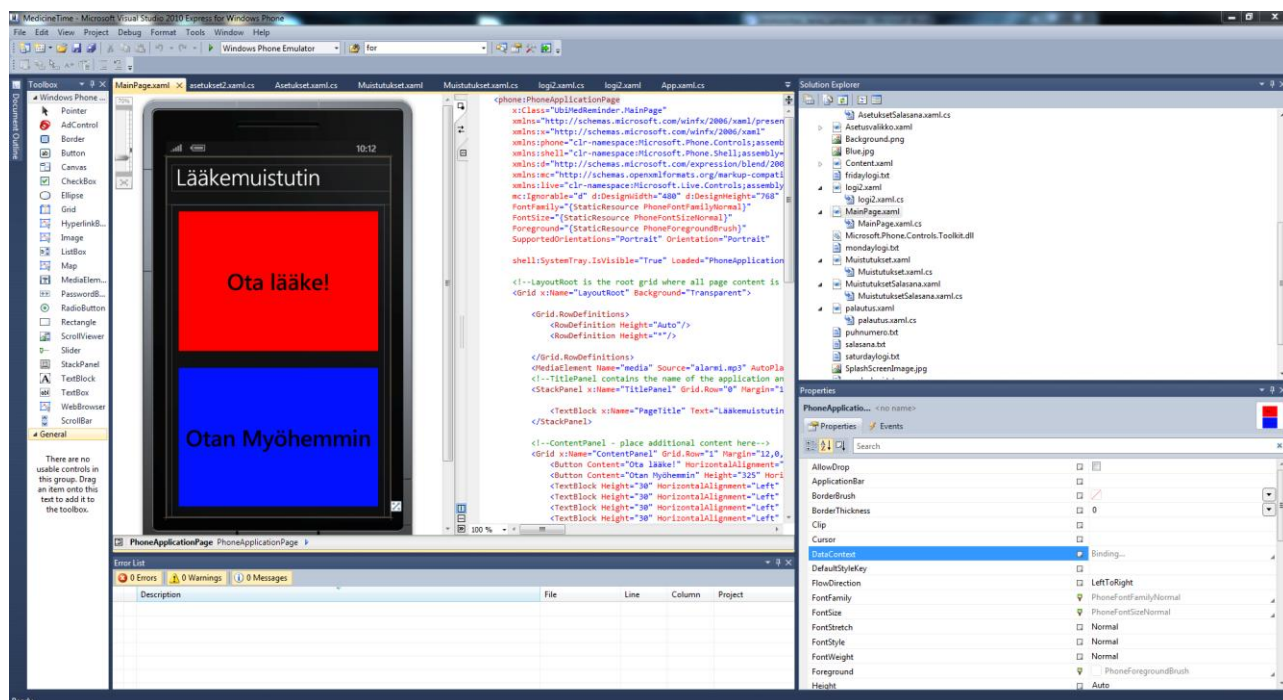
XAML tulee sanoista Extensible Application Markup Language, ja sitä käytetään .NET-sovelluskehityksen ohjelmointimallissa selventämään käyttöliittymän suunnittelua .NET-ohjelmointikieleen pohjautuvissa sovelluksissa. Windows Phone -sovellusten suunnittelussa XAML on tärkeässä osassa käyttöliittymän suunnittelua(What is XAML 2012, hakupäivä 5.5.2012.)

3.2 Windows Live SkyDrive

Windows Live SkyDrive on Microsoftin kehittämä pilvipalvelu, johon saa maksutta vapaata tallennustilaa 25 GB, jossa rajoituksena suurin mahdollinen tiedostokoko on 2 GB. Palvelu on rakennettu HTML5-teknologiaa käyttämällä ja se on erittäin helppokäyttöinen. Palvelun käyttäminen vaatii Windows Live tunnukset (SkyDrive 2012, hakupäivä 10.4.2012.) Tutkin mahdollisuutta lisätä SkyDrive -ominaisuuden Medicine Time -sovellukseen tässä opinnäytetyössä.

3.3 Microsoft Visual Studio 2010

Microsoft Visual Studio 2010 on Microsoftin vuonna 2010 julkaisema ohjelmointiympäristö(IDE). Tässä versiossa on aikaisempiin nähden yksinkertaistettu valikoita helppokäyttöisemmiksi, sekä parannettu help-toimintoa, jolla käyttäjät saavat apua sovelluksen käytössä (Wikipedia 2012,Hakupäivä 6.5.2012.) Kuvassa 3 nähdään Visual Studion käyttöliittymä.



KUVA 3: Microsoft Visual Studio 2010 -sovelluksen käyttöliittymä

Kehitin Medicine Time -sovelluksen Microsoft Visual Studiolla. Visual studio sisältää emulaattorin, jolla voidaan simuloida Windows Phone -matkapuhelinta tietokoneen ruudulla. Emulaattori on hyvä työkalu nopeaan toiminnallisuuden testaamiseen, mutta se ei anna oikeaa kuvaa käytettävyydestä, ja lisäksi se sisältää rajoituksia, kuten se, että sammutettaessa emulaattori poistaa kaikki tallennetut tiedot. Näin ollen lokitoimintoa ei pelkällä emulaattorilla voida testata.

3.4 Nokia Lumia

Jotta kaikkia ominaisuuksia voitiin testata, tarvittiin emulaattorin sijasta oikea Windows Phone 7.5 -alustaa käyttävä matkapuhelin. Tätä varten sain Oulun seudun ammattikorkeakoululta käyttööni Nokia Lumia 800 -matkapuhelimen(kuva 4). Nokia Lumia -älypuhelimet edustavat tämän hetken Windows Phone -alustaa käyttävien puhelimien kärkeä, ja näin ollen ne ovat varsin sopivia sovelluksen testaamiseen. Ostin myös itselleni Nokia Lumia 710 -matkapuhelimen (kuva 5), jota käytin myös testaamiseen.



KUVA 4: Nokia Lumia 800(Nokia Lumia 800, hakupäivä 4.5.2012)



KUVA 5: Nokia Lumia 710(Nokia Lumia 710, hakupäivä 4.5.2012)

3.5 Microsoft Expression Blend 4

Microsoft Expression Blend on Microsoftin kehittämä käyttöliittymän suunnittelutyökalu. Microsoft Expression Blendin versio 4 julkaistiin vuonna 2010, ja sen Windows Phone -alustalle tarkoitettu versio on maksuton, ja löytyy valmiiksi Windows Phone kehittäjätyökalupaketista. Kuvassa 6 nähdään Microsoft Expression Blend 4 -sovelluksen käyttöliittymä. Käytin Microsoft Expression Blend 4-sovellusta käyttöliittymän hienosäätämiseen, kun toteutus oli muuten valmis.



KUVA 6: Microsoft Expression Blend 4 -sovelluksen käyttöliittymä

4 SOVELLUKSEN TOTEUTUS

Ensimmäinen versio työstä on itsenäinen sovellus, jonka kuka tahansa voi ostaa Windows Phone Marketplacesta. Sovelluksen tarkoituksena on toimia ubiPILL-järjestelmän hälytinlaitteen erillISRatkaisuna. Tärkein ero aikaisempaan järjestelmään on se, että sovellus ei tarvitse useasta erillisesti osasta koostuvaa järjestelmää toimiakseen, vaan on oma järjestelmä itsessään.

Työn toteuttaminen oli mielenkiintoista, sillä Windows Phone on alustana hyvin ajankohtainen ja tuore. Heti alusta asti minulla oli selkeä visio, millainen sovelluksesta pitäisi tulla, mutta toteutustavat eivät olleet täysin selvillä alussa.

Seurasin vesiputousmallia vapaasti mukaillen sovellusta kehittäessäni helpottaakseni kokonaisuuden hahmottamista. Jaoin järjestelmän eri moduuleihin, joita kehitin ja testasin yksi kerrallaan, kunnes ne olivat määritykset täyttävällä tasolla.

Sovelluksen kehittäminen ei ollut läheskään ongelmaton. Koska Windows Phone on alustana vielä nuori, ovat monet koodikirjastot puutteellisia, sekä bugeja ilmenee silloin tällöin. Lisäksi Microsoft itse valvoo sekä rajoittaa alustan muokkausta, joka saattaa rajoittaa erinäisten toimintojen ohjelmoimista.

Aluksi kerron, kuinka Windows Phone -alustan sovelluksen kehittämisen vaiheista, ja sitten kuvailen kuinka Medicine Time -lääkemuistutin on toteutettu.

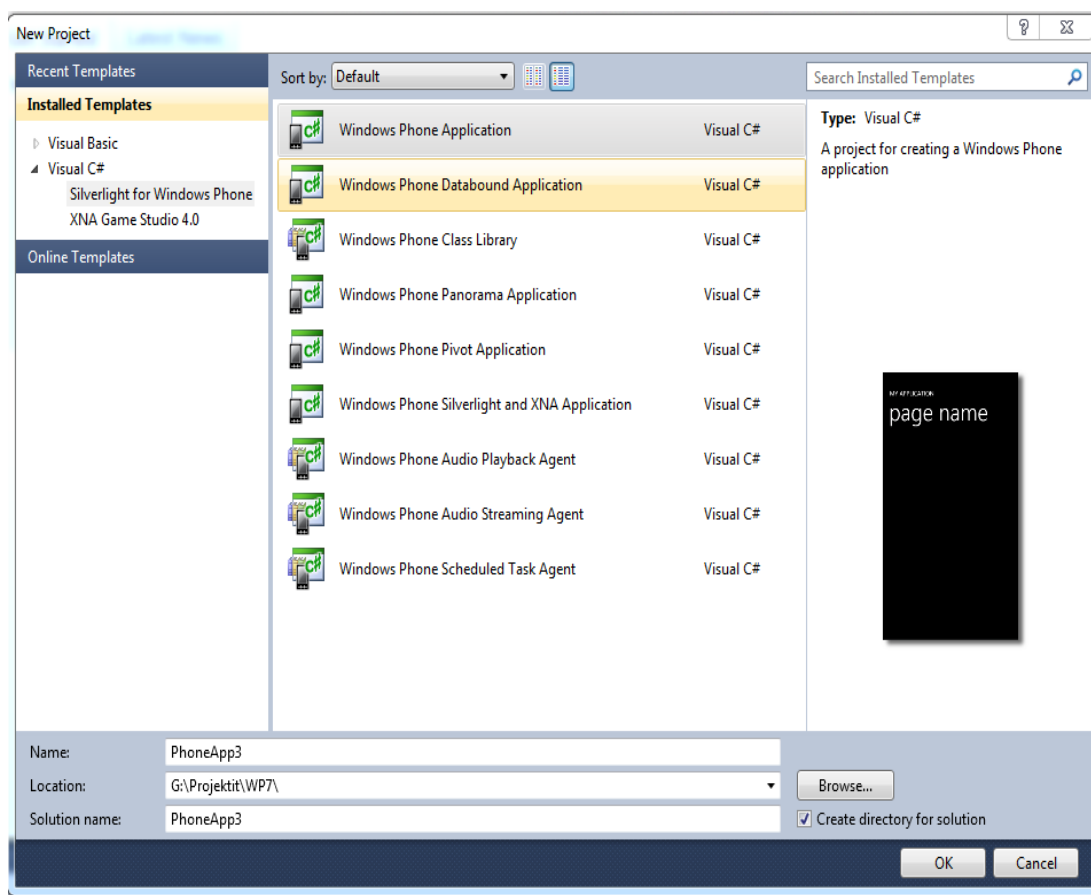
4.1 Windows Phone sovelluksen kehittäminen Visual Studiolla

Kun uutta projektia aletaan kehittää Windows Phone -alustalle, valitaan projektille pohja erilaisista projektimalleista. Aluksi on valittava kahden ohjelmointikielen väliltä, jotka ovat Visual Basic ja Visual C#. Itse valitsin sovellukselleni Visual C# -ohjelmointikielen nykyaikaisempien koodikirjastojen

vuoksi ja siksi, että se on läheistä sukua c-ohjelmointikielelle, josta minulla on aikaisempaa kokemusta.

Kun ohjelmointikieli on päätetty, valitaan, tehdäänkö sovellus käyttäen hyväksi Silverlight-ohjelmistokehystä vai XNA-ohjelmistokehystä. XNA on tarkoitettu lähinnä pelien kehittämistä varten, joten valitsin Silverlight-pohjaisen sovelluksen.

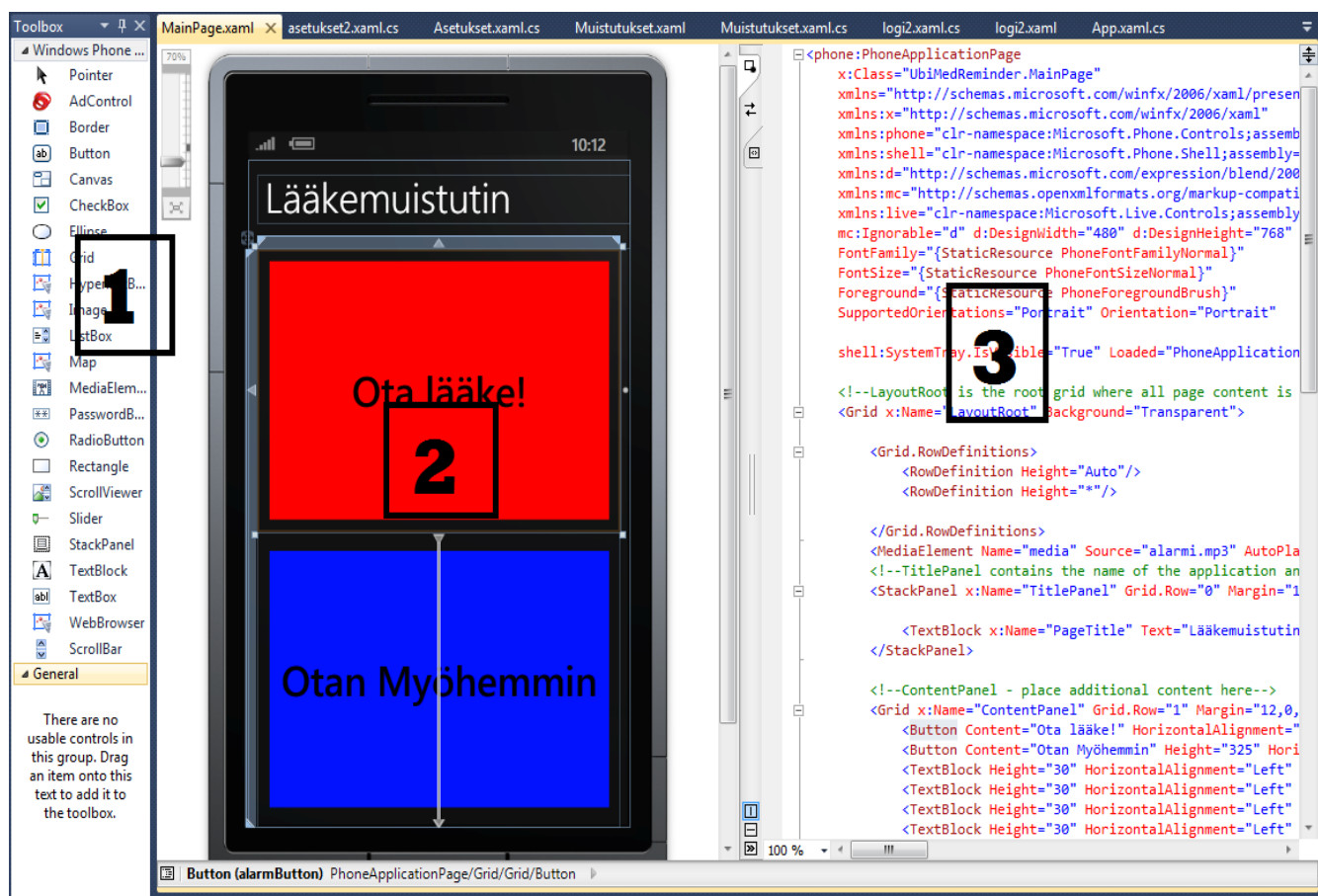
Jokainen Silverlight-pohjainen projekti koostuu useista luokista, jotka on jaettu kahteen osaan: XAML-ohjelmointikielellä toimivaan käyttöliittymäosaan sekä Visual Basic- tai Visual C# -ohjelmointikieleen perustuvaan luokkaosaan, jossa toteutetaan sovelluksen toiminnallisuus. Valitsin Medicine Time -sovellukselle "Windows Phone Application" -projektimallin, joka on kuvassa 7.



KUVA 7: Visual Studio 2010 -sovelluksen projektimallin valinta

4.1.1 Käyttöliittymän suunnittelu

Kun projektimalli on valittu, aletaan suunnitella sovellukselle käyttöliittymää. Käyttöliittymän suunnitteluun Microsoft Visual Studio 2010 -sovelluksesta löytyy omat työkalut, joilla objektien lisääminen ja muokkaaminen käyttöliittymään on tehty helpoksi ja yksinkertaiseksi. Kuvassa 8 nähdään Visual studio 2010 -sovelluksen käyttöliittymäsuunnitteluun käytettävä työkalu.



KUVA 8: Visual studio 2010 -sovelluksen käyttöliittymän suunnittelu työkalu

Kuvan kohdassa 1 voidaan valita joku objekteista, joka voidaan viedä vetämällä sovelluksen käyttöliittymään kohdassa 2.

Kohdassa 2 käyttöliittymään on viety kaksi nappia, joiden ulkonäköä on muutettu vastaamaan määritysten mukaisia vaatimuksia. Ulkonäköä voidaan muokata joko Visual Studiosta löytyvillä työkaluilla, tai Expression Blend 4 -sovelluksella, joka on tarkoitettu käyttöliittymän ulkoasun hiomiseen.

Kohdassa 3 on käyttöliittymän tiedot XAML-ohjelmointikielenä. XAML on kieli, jonka tarkoituksena on selventää käyttöliittymän suunnittelua esittämällä objektit ja niiden ominaisuudet tekstimuodossa. Kuvassa 9 on esitelty esimerkin kautta XAML-koodia.

```
<!--ContentPanel - place additional content here-->
<Grid x:Name="ContentPanel" Grid.Row="1" Margin="12,0,12,0">
    <Button Content="Ota lääke!" HorizontalAlignment="Left" Margin="0,0,0,339" Name="alarmButton" Width="456" Backgrou
    <Button Content="Otan Myöhemmin" Height="325" HorizontalAlignment="Left" Margin="0,333,0,0" Name="sleepButton" Ver
    <TextBlock Height="30" HorizontalAlignment="Left" Margin="186,262,0,0" Name="outputBlock" Text="" VerticalAlignmer
    <TextBlock Height="30" HorizontalAlignment="Left" Margin="139,280,0,0" Name="textBlock1" Text="" VerticalAlignment
    <TextBlock Height="30" HorizontalAlignment="Left" Margin="311,325,0,0" Name="textBlock2" Text="" VerticalAlignment
    <TextBlock Height="30" HorizontalAlignment="Left" Margin="311,329,0,0" Name="contentBlock" Text="" VerticalAlignme
</Grid>
</Grid>
```

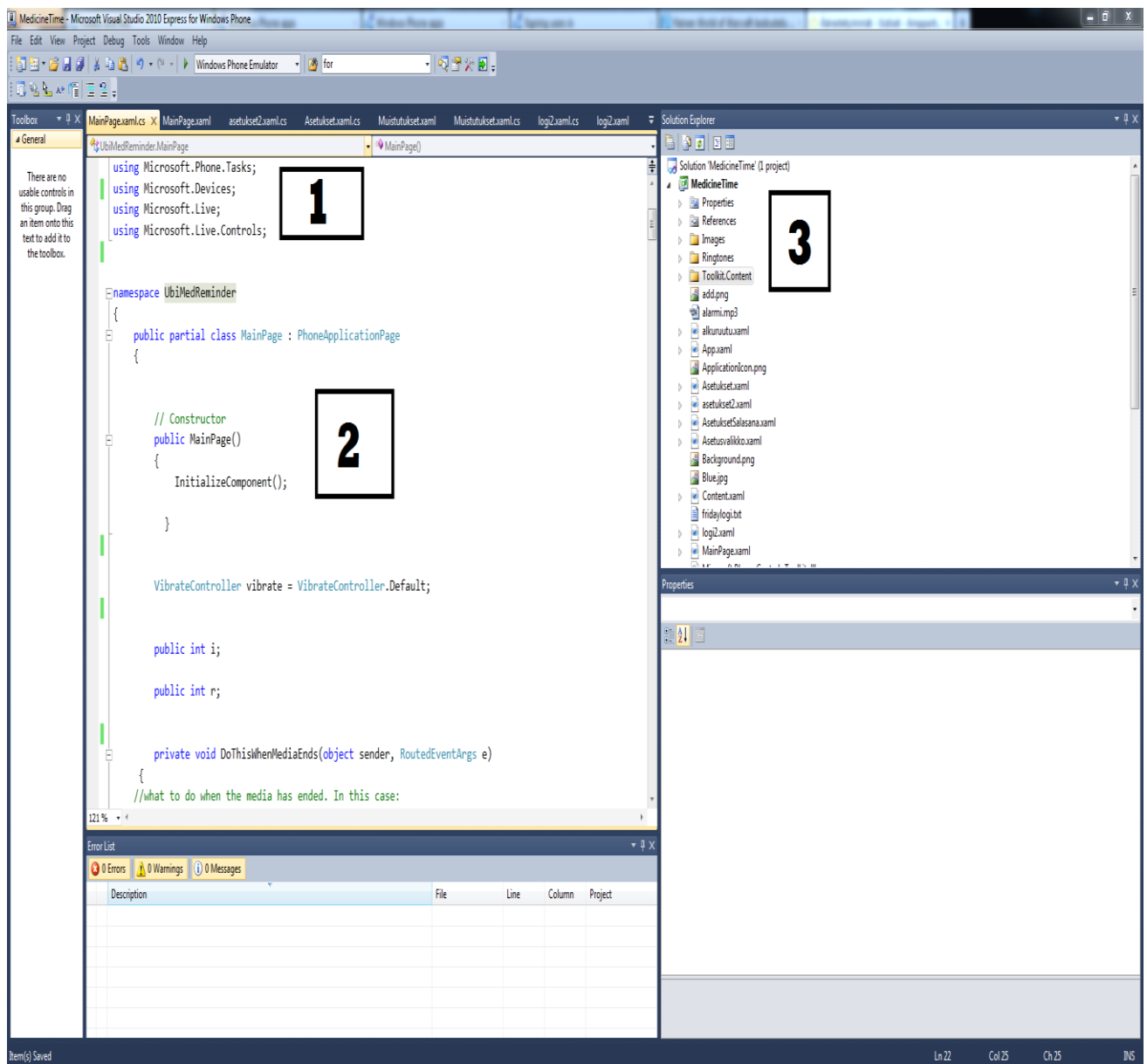
KUVA 9: Esimerkki XAML-koodista.

Koodissa on määritelty ulkoasuksi Grid, jonka sisälle menevät kaksi painiketta, joiden ominaisuudet on määritelty koodissa. Esimerkiksi kohdassa Button Content on määritelty painikkeessa lukeva teksti, HorizontalAlignment sekä Margin määrittävät painikkeiden paikat sekä Name-kohdassa on määritelty nimi, jolla viitataan painikkeisiin toiminnallisuutta kehitettäessä.

TextBlock on objekti, joka voi sisältää tekstiä, ja se voi olla joko näkyvä tai piilotettu objekti. Tässä tapauksessa koodista löytyvät 4 TextBlock-objektia ovat piilotettuja, ja niitä käytetään ottamaan vastaan tietoa esim. kellonajasta, jolloin lääke on otettu.

4.1.2 Toiminnallisuuden suunnittelu

Kun sovelluksen käyttöliittymän ensimmäinen testiversio on valmis, voidaan toiminnallisuutta alkaa kehittää. Toiminnallinen osuus kehitetään käyttäen valittua ohjelmointikieltä Visual studio -sovelluksesta löytyvillä kehitystyökaluilla. Kuvassa 10 nähdään kehitystyökalu sovelluksen toiminnallisuuden kehittämiseen.



Kuva 10: Visual Studio 2010 -sovelluksen kehitystyökalut Visual C# ohjelmointikielelle

Kuvan kohdassa 1 on määritelty sovelluksen käyttämät kirjastot. Jokaisella luokalla on omat määritetyt kirjastot joita sovellus käyttää. Esimerkiksi hälytintä suunniteltaessa, täytyy Microsoft.Devices-kirjasto määritellä, jotta värinän käyttö voidaan ohjelmoida hälyttimeen.

Kuvan kohdassa 2 määritetään sovelluksen toiminnallisuus Visual C# -ohjelmointikielenä. Alla esimerkki Visual C# -ohjelmointikielestä.

```
private void sleepButton_Click(object sender, RoutedEventArgs e)
{
    media.Stop();
    vibrate.Stop();
}
```

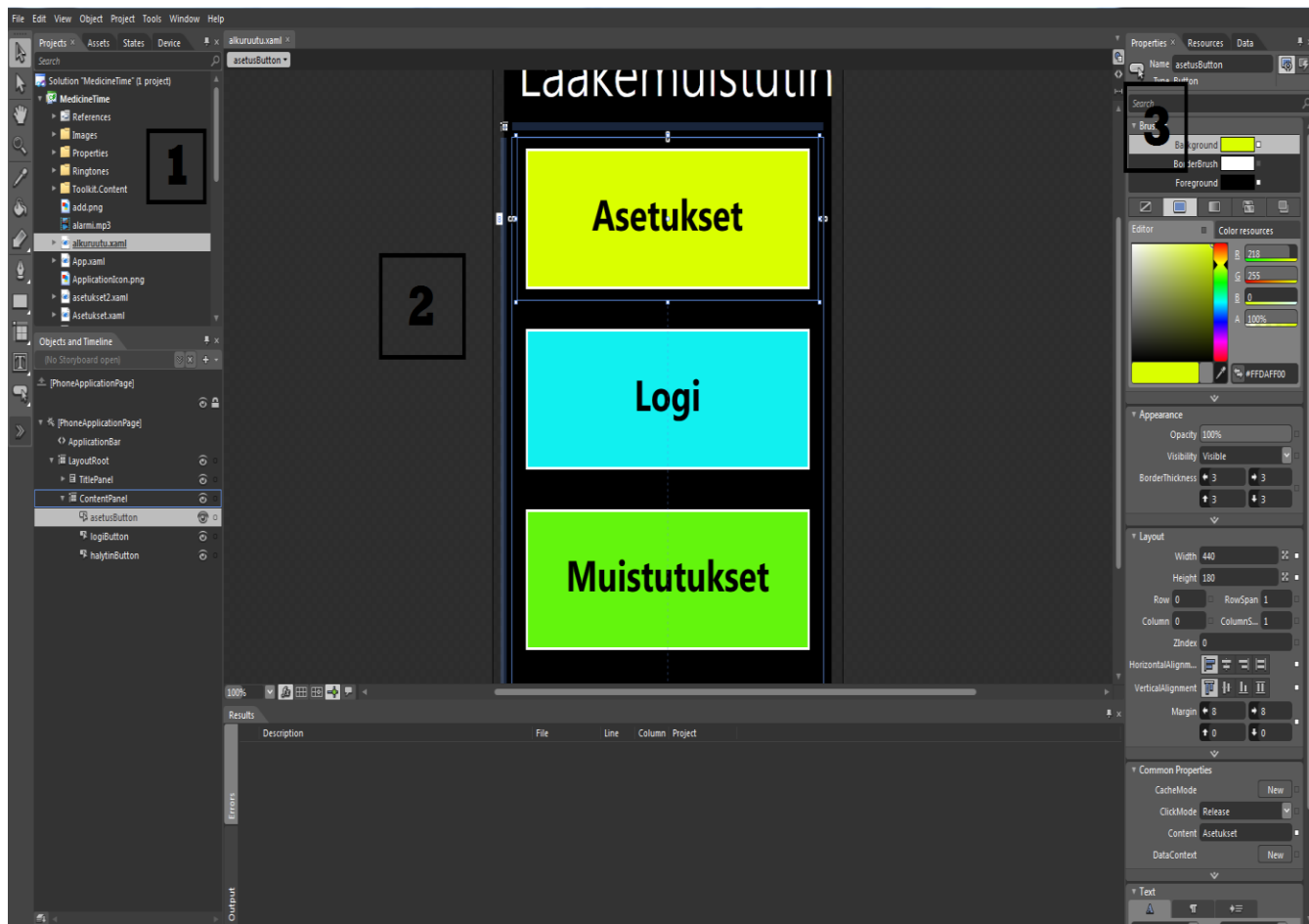
ESIMERKKI 1: sleepButton-painikkeen toteutus Visual C# -ohjelmointikielellä

Esimerkissä 1 sleepButton-nimisen painikkeen painallus toteuttaa komennot media.Stop, joka lopettaa hälytysäänien soittamisen, ja vibrate.stop, joka lopettaa värinähälytyksen.

Kuvan 10 kohdassa 3 näkyy navigaatioalue, jossa näkyvät kaikki osat, joista projekti koostuu ja joita voidaan valita muokattavaksi kehitystyökalussa.

4.1.3 Käyttöliittymän viimeistely

Kun sovelluksen toiminnallisuus on toteutettu ja testattu, tehdään viimeiset säädöt käyttöliittymään Microsoft Expression Blend 4 -sovelluksella. Toteutettu projekti ladataan Expression Blend -sovellukseen, ja sen jälkeen valitaan projektin osa, jonka ulkoasua halutaan muokata. Kuvassa 11 nähdään Expression Blend 4, johon on ladattu Medicine Time -sovellus.



KUVA 11: Microsoft Expression Blend 4 -sovelluksen käyttöliittymän suunnittelutyökalut.

Kuvan kohdassa 1 näkyvät ladatun projektin eri osat. Tässä kuvassa olen valinnut alkuruutu-nimisen sivun muokattavaksi.

Kuvan kohdassa 2 on valitun sivun käyttöliittymä, jota voidaan muokata suoraan esim. muuttamalla painikkeiden kokoa venyttämällä niitä suoraan nurkista tai muuttamalla kuvan kohdassa 3 objektien ominaisuuksia.

Kuvan kohdassa 3 nähdään objektien ominaisuudet. Tässä kuvassa olen valinnut keltaisen asetuspainikkeen nimeltä asetusButton ja näin ollen sen ominaisuuksia voidaan muuttaa tällä sovelluksella. Yleisiä painikkeen muokattavia ominaisuuksia ovat esimerkiksi sen väri, pituus ja leveys, painikkeen sijainti käyttöliittymässä sekä painikkeen tekstin ominaisuudet.

4.1.4 Sovelluksen lisääminen Windows Phone Marketplace -palveluun

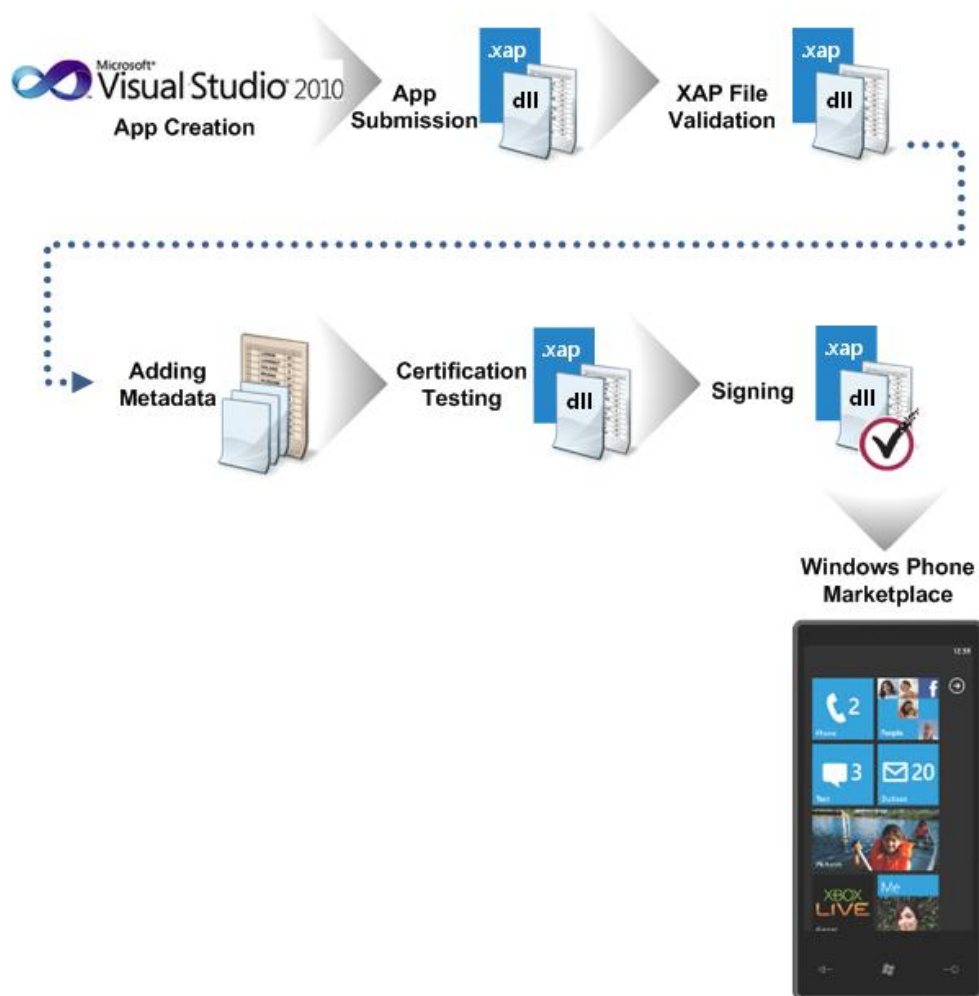
Valmis sovellus laitetaan myyntiin Windows Phone Marketplace -palveluun. Visual Studio -sovelluksesta löytyvillä työkaluilla voidaan ensin testata sovelluksen kelpoisuus Marketplace-palveluun. Työkalut sisältävät testit esimerkiksi sille, ovatko sovelluksen muistinkäyttö ja käynnistymisajat vaatimukset täyttäviä. Myös sovelluksen sisäistä navigaatiota ja kaatumattomuutta voidaan testata.

Jotta sovelluksia voitaisiin laittaa myyntiin Marketplaceen, on Microsoft asettanut niille seuraavat vaatimukset:

- 1.Sovellukset ovat luotettavia
- 2.Sovellukset käyttävät resursseja tehokkaasti.
- 3.Sovellukset eivät ole ristiriidassa puhelimen käytettävyyden kanssa.
- 4.Sovellukset eivät sisällä viruksia tai muita haittaohjelmia.

(Application Certification Requirements for Windows Phone 2012, hakupäivä 12.5.2012.)

Jos sovellus täyttää edellä mainitut vaatimukset, voidaan sovellus lähettää Microsoftin tarkastettavaksi App Hub -sivuston kautta. Microsoft tarkistaa sovelluksen kuvassa 12 näkyvän prosessin mukaisesti. Microsoftin tarkistuksessa menee yleensä 2-5 päivää ja, jos virheitä havaitaan, saa lähettäjä virheraportin.



KUVA 12: Sovelluksen lähettäminen Marketplace -palveluun(Application Certification Requirements for windows Phone 2012, hakupäivä 12.5.2012)

Sovellusta lähettäessä pitää myös antaa sovelluksesta kuvankaappauksia sekä kuvat kuvakkeille, millaisina sovellus Marketissa ja puhelimessa nähdään. Lisäksi sovellus pitää kuvailla sanoin sekä asettaa myyntihinta. Myyntihintaa asettaessa sovellukselle määritetään myös myynti-alue joko maanosan tai maan mukaan (Application Certification Requirements for windows Phone,2012, hakupäivä 12.5.2012).

4.2 Medicine Time -sovelluksen toteutus

Aloitin Medicine Time -sovelluksen kehittämisen kehittämällä yksinkertaisen testiversion käyttöliittymästä. Tämä testiversio nähdään kuvassa 13.



KUVA 13 : Ensimmäinen versio käyttöliittymästä

Kun käyttöliittymän kuvan 13 mukainen versio oli valmis, aloin toteuttaa järjestelmän eri osioita. Jaoin suunnitteluvaiheessa järjestelmän neljään eri osaan: hälytinsosaan, lokikirjaosaan, asetusosaan sekä muistutinosaan. Näistä aloin tarkastella miten hälytinosan voisi toteuttaa.

4.2.1 Hälyttimen toiminnallisuuden suunnittelu

Aloitin sovelluksen toiminnallisuuden kehittämisen hälytinosasta. Määritysten mukaisesti hälytinosi koostuisi kahdesta napista, joista toisella käyttäjä kuittaa ottaneensa lääkkeen ja toisella siirtää hälytystä myöhemmäksi.

Äänen toteuttaminen

Aluksi selvitin, miten hälyttimeen saataisiin ääni, joka lopettaisi soimisen, kun käyttäjä kuittaa hälytyksen jommallakummalla napilla. Ratkaisu löytyi hetken etsimisen jälkeen Microsoftin omilta MSDN-sivuilta, joista löytyi ratkaisuksi upottaa käyttöliittymään XAML-ohjelmointikielellä MediaElement-ominaisuus. MediaElement alkaa toistaa ääntä, kun käyttäjä navigoi sovelluksessa sivulle, jossa kyseinen ominaisuus löytyy. Äänen toiston lopettamista varten käytin media.stop -komentoa, jonka lisäsin kumpaankin painikkeeseen (Media Element Class 2012, hakupäivä 12.5.2012.)

Värinän toteuttaminen

Kun hälyttimen ääni oli saatu toteutettua, tutkin, miten puhelimen saisi värisemään hälytyksen yhteydessä. Ratkaisu löytyi myös MSDN-sivulta, josta sain tietää, että värinän hallintatyökalut löytyvät Microsoft.Devices-koodikirjastosta. Vibrate.Start-komento aloittaa värinän, ja se voidaan määrittää värisemään enintään 5 sekuntia yhtäjaksoisesti. Koska monen sekunnin toistuva värinä olisi hyödyllisyyden sijasta vain epämukava, halusin värinän jaksottaiseksi (Microsoft.Devices Namespace 2012, hakupäivä 12.5.2012.)

Toteutuksen tähän löysin e-kirjasta Programming Windows Phone 7. Kirjassa on esimerkki yksinkertaisesta ajastinsovelluksesta. Tässä esimerkkitsovelluksessa käytetään hyödyksi DispatcherTimer-ominaisuutta, jonka sisälle voidaan ajastaa asioita tapahtumaan tietyn ajanjakson välein (Petzold, 2010, 57).

Toteutin värinän käyttäen hyödyksi tätä ajastinta sekä While-silmukkaa. While-silmukka toteuttaa sen sisään ohjelmoitua käskyä niin kauan, kunnes sen

määrittelyssä oleva ehto muuttuu. Laitoin While-silmukan pyörimään niin kauan, kunnes toista painikkeista on painettu ja ajastimen pyörimään silmukan sisään joka sekunnin välein. Joka sekunti ajastin käynnistää värinän Vibrate.Start-komennolla puoleksi sekunniksi, jolloin värinä tulee jaksottaisesti puolen sekunnin välein.

Lokikirjoituksen toteuttaminen

Seuraavaksi aloin perehtyä, kuinka toteuttaa merkinnät lokikirjaan siitä, milloin käyttäjä on ottanut lääkkeen, tai lääke on jäänyt ottamatta. Aluksi tutustuin Windows Phonen käyttämään tietojen tallennustapaan. Lukemalla Programming Windows Phone 7 -kirjaa selvisi, että Windows Phone 7 -sovellukset käyttävät tiedon tallentamiseen Isolated Storage -nimistä tallennustilaa. Isolated Storage toimii siten, että kaikki sovellukset pääsevät käsiksi tähän tallennustilaan, mutta vain sovellukset, jotka ovat luoneet siellä sijaitsevat tiedostot, voivat käyttää niitä. Näin ollen tiedot pysyvät piilossa, eivätkä muut sovellukset pääse hyväksikäyttämään esimerkiksi Medicine Time -sovelluksen lokitietoja(Petzold 2010, 134.)

WindowsPhoneGeek -sivustolta löysin apua siihen, miten pystyisin tallentamaan tekstitiedostoja käyttäen Isolated Storage -tallennustilaa.

Jotta sovellus voisi käyttää tiedon tallentamiseen tarkoitettuja komentoja, täytyy ottaa käyttöön System.IO- ja System.IO.IsolatedStorage-koodikirjastot. (All About WP7 Isolated Storage - Read and Write text files, 2012, hakupäivä 1.2.2012).

Tämän jälkeen käyttäen StreamWriter-ominaisuutta voidaan tekstiä kirjoittaa haluttuun tiedostoon (StreamWriter Class 2012, hakupäivä 12.5.2012). Tein jokaiselle päivälle oman tekstitiedoston, jolloin samana viikonpäivänä otetut lääkkeet menevät samaan tekstitiedostoon. Tämä selkeyttää lokien lukemista, kun voidaan tarkastella jokaista viikonpäivää erikseen.

Kun olin perehtynyt tietojen kirjoittamiseen, aloin tutkia, miten järjestelmän kellosta saadaan otettua aika lokikirjaa varten. Vastauksen tähänkin löysin

Programming Windows Phone 7 -kirjasta. Siellä olevasta esimerkissä HybridClock käytettiin DateTime-luokkaa tulostamaan senhetkinen aika (Petzold,2010, s189). Tutkimalla DateTime-luokkaa MSDN-sivulta löysin sen ominaisuuksista mahdollisuuden tallentaa sen hetkisen ajan String-muuttujaksi, joka voitaisiin kirjoittaa lokiin StreamWriterin avulla. (DateTime Structure, 2012, hakupäivä 12.5.2012).

Löysin myös DateTime-luokasta ominaisuuden, jolla saadaan lääkkeenottohetken viikonpäivä selville. Tätä sekä IF-komentoa hyödyntäen sain määritettyä ohjelmalle sen, minkä päivän lokitiedostoon tiedot tallennetaan.

IF-komento toimii siten, että IF-komennolle asetetaan ehto, ja ehdon täyttyessä toteutetaan sen sisällä oleva silmukka. Tein jokaiselle viikonpäivälle oman silmukan, josta IF-komento karsii lokitiedon oikealle päivälle.

Tämän jälkeen siirryin toteuttamaan lokikirjaosaa, jotta näkisin, että tekstitiedostoihin kirjoittaminen varmasti toimii oikein.

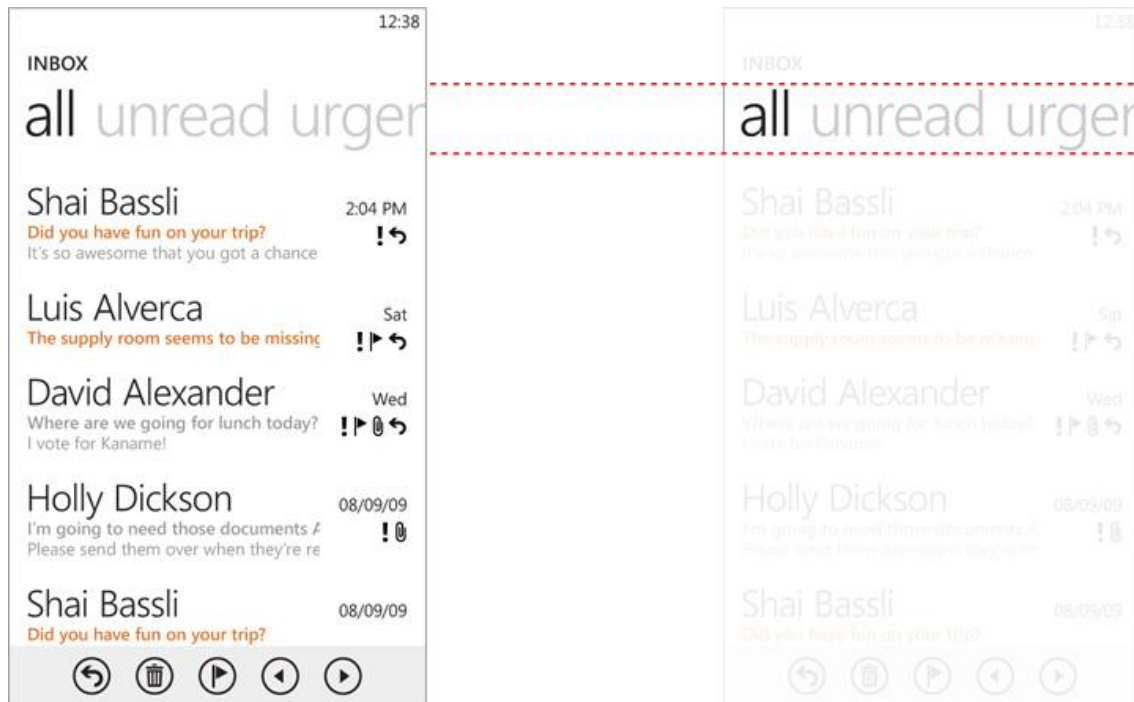
4.2.2 Lokikirjan toiminnallisuuden suunnittelu

Määritysten mukaisesti lokikirjassa täytyisi näkyä selkeästi, milloin lääkkeitä on otettu ja jäänyt ottamatta. Myös käyttäjällä pitäisi olla mahdollisuus tarkistaa, milloin hän on ottanut lääkkeit, sekä omaisella täytyisi olla mahdollisuus myös tyhjentää lokitietoja.

Lokikirjan ulkoasu

Lokikirjan toteutusta varten mietin eri vaihtoehtoja, miten lokista saataisiin mahdollisimman selkeä ja helppokäyttöinen. Tutkimalla MSDN-sivuston harjoituksia löysin ratkaisuksi toteuttaa lokikirjan Pivot-sivuna (Pivot Control Overview 2012, hakupäivä 12.5.2012).

Pivot-sivun ylälaitaan voit lisätä uusia välilehtiä, joita voidaan selata pyyhkäisemällä sormella joko vasemmalle tai oikealle, jolloin sivuun avautuu uusi välilehti. Kuvassa 14 on kuvailtu Pivot-sivun välilehtien rakenne.



KUVA 14: Pivot-sivu (Pivot Control Overview for Windows Phone 2012, hakupäivä 12.5.2012)

Toteutin lokikirjan selauksen seitsemänä eri välilehtenä, jotka on nimetty viikonpäivien mukaan. Tällöin navigointi halutun viikonpäivän lokitietoihin on helppoa. Koska hälytinsovellus kirjoittaa lääkkeenottoajan tekstitiedostoon käyttäen StreamWriteriä, löytyy tekstitiedostojen lukemiselle myös oma StreamReader-luokka, jonka avulla muistiin kirjoitetut lokitiedot saadaan tulostettua lokikirjaan(StreamReader Class, 2012, hakupäivä 12.5.2012). Myös käyttäjä voi tarkastella lokitietoja, mutta vain omainen voi tyhjentää lokikirjaa, jos hän valitsee Muokkaa lokia -painikkeen asetusvalikosta.

4.2.3 Asetusten toiminnallisuuden suunnittelu

Kun hälytin ja lokikirja oli toteutettu, siirryin suunnittelemaan asetusten toteutusta. Jotta puhelimen asetuksia pääsisi säätämään ainoastaan omainen, täytyi asetukset suojata salasanalla. Aluksi tein sivun, johon käyttäjä voi asettaa haluamansa salasanan PasswordBox-nimiseen kenttään. PasswordBox-kenttä salaa tekstin korvaamalla kentässä näkyvän tekstin *-merkillä. Kun käyttäjä painaa tallenna-painiketta, StreamWriter-ominaisuus tallennetaan salasanan tekstitiedostoon, jossa se pysyy tallessa.

Seuraavaksi tein sivun, joka pyytää antamaan salasanan, jotta asetuksia päästään muokkaamaan. Salasanan tarkastaminen on toteutettu käyttäen StreamReader-ominaisuutta. Kun käyttäjä painaa OK-painiketta, lukee StreamReader salasana-tekstitiedoston ja vertailee annettua salasanaa. Jos salasana on väärä, ilmoittaa sovellus siitä tekstillä "Väärä salasana!". Jos taas salasana on oikein, aukaisee sovellus asetussivun.

Vakiona salasana on tyhjä, joten omaisen on suositeltavaa asettaa sovellukselle salasana, ennen kuin sovellus otetaan käyttöön.

Myöhemmin lisäsin mahdollisuuden palauttaa salasana, jos se on unohtunut. Lisäsin asetussivulle kaksi tekstikenttää, joihin voidaan asettaa turvakysymys ja turvavastaus. Suositeltavaa on, että kysymys on sellainen, johon vain kysymyksen asettaja tietää vastauksen.

Salasanan palauttaminen tapahtuu painamalla salanasivulla palautus-painiketta. Avautuva sivu kysyy turvakysymyksen ja, jos siihen vastaa oikein, kertoo sovellus oikean salasanan.

Muita myöhemmin lisättyjä ominaisuuksia asetuksiin ovat lokikirjan muokkaaminen sekä toisen PasswordBox-kentän lisääminen, jotta kun käyttäjä

haluaa vaihtaa salasanaa, täytyy se kirjoittaa kahteen kertaan, ettei vahingossa pääse tallentumaan salasana jossa, on kirjoitusvirheitä.

4.2.4 Muistutusten toiminnallisuuden suunnittelu

Viimeisenä aloin selvittämään, miten muistutusosa voitaisiin toteuttaa. Aluksi tutkin, voidaanko muistutukset toteuttaa käyttäen taustaohjelmia.

Muistuttimen toteuttaminen

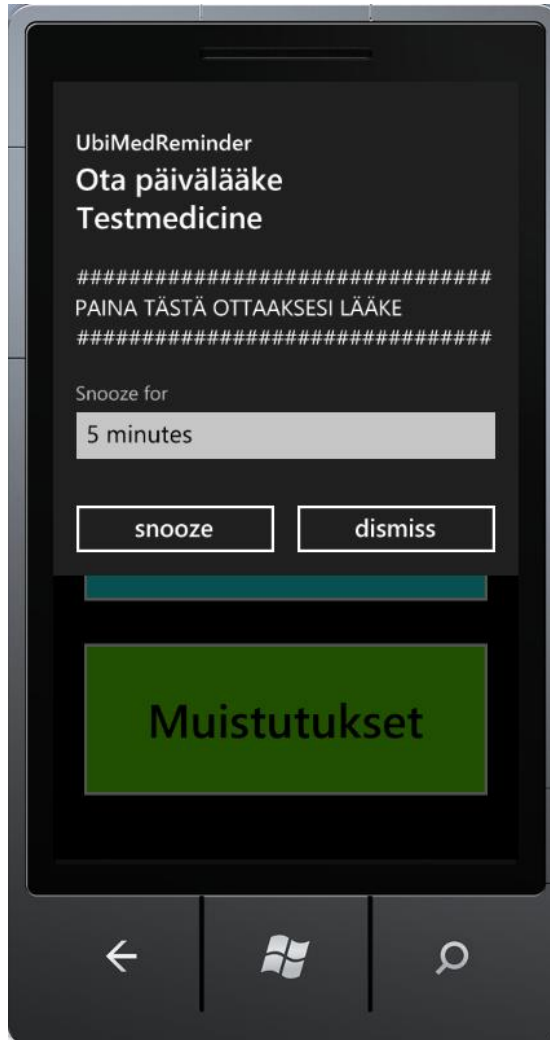
Windows Phone 7.5 alusta tukee kahta erilaista taustasovellustyyppiä, `PeriodicTask` sekä `ResourceIntensiveTask`. `PeriodicTask`-taustaohjelmassa sovelluskirjastojen käyttöä on rajoitettu ja taustaohjelmia voidaan ajaa vain kerran puolessa tunnissa, jolloin muistutusta ei voida rakentaa tätä tehtävää hyväksikäyttäen (Background Agents Overview for Windows Phone 2012, hakupäivä 12.2.2012).

`ResourceIntensiveTask`-taustaohjelmaa voidaan käyttää vain, jos puhelimen tila täyttää tietyt vaatimukset liittyen prosessorin aktiivisuuteen, virtalähteeseen, sekä internetyhteyteen. Tämän takia tätäkään taustaohjelmatyyppiä ei voitu käyttää muistutuksien toteuttamiseen. (Background Agents Overview for Windows Phone 2012, hakupäivä 12.2.2012).

Koska taustaohjelmia ei voitu käyttää muistutusten toteuttamiseen, tutkin, voisiko puhelimen omaa sisäistä hälytin- ja muistutin-toimintoa käyttää oman Medicine Timen toteuttamiseen. Windows Phonen käyttämä Scheduler-kirjasto sisältää kaksi hälytinsovellusta: `Alarm` ja `Reminder` (`Microsoft.Phone.Scheduler.Namespace`, 2012, hakupäivä 12.5.2012).

`Alarm`-toiminto on Windows Phonen sisäinen hälytin, jolle voidaan asettaa oma soittoääni sekä ajastus, mutta sitä käyttämällä ei voida kuitenkaan käynnistää uusia sovelluksia. `Reminder`-toiminto on Windows Phonen -muistutinohjelma, jonka soittoääntä ei voi vaihtaa, mutta sen kautta voidaan aukaista uusi

sovellus. Toteutin muistutukset käyttämällä tätä Reminder-luokkaa. Kun ajastettu muistutus käynnistyy, avautuu kuvan 14 mukainen näkymä.



KUVA 14: Reminder-toiminnon ruutuun pomppaava ikkuna

Jos käyttäjä painaa tekstiä "PAINA TÄSTÄ OTTAAKSESI LÄÄKE", aukeaa varsinainen hälytinsovellus(kuva 15). "Snooze for" -kohdasta voidaan asettaa haluttu torkku aika hälyttimelle, milloin se hälyttää uudelleen, jos käyttäjä ei ole kotona tai dosetin lähellä.

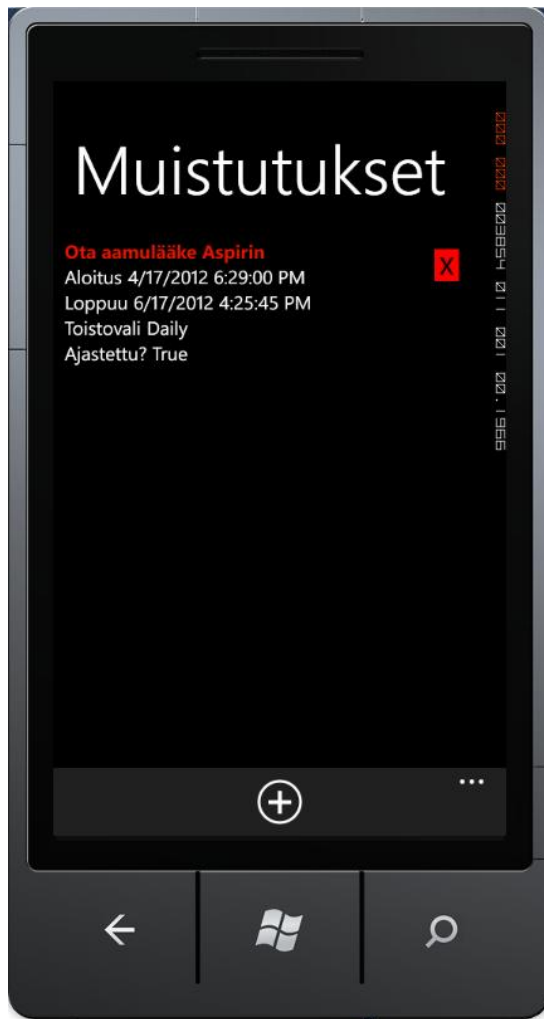


KUVA 15: Hälytinsovellus

Uusien muistutusten asettaminen

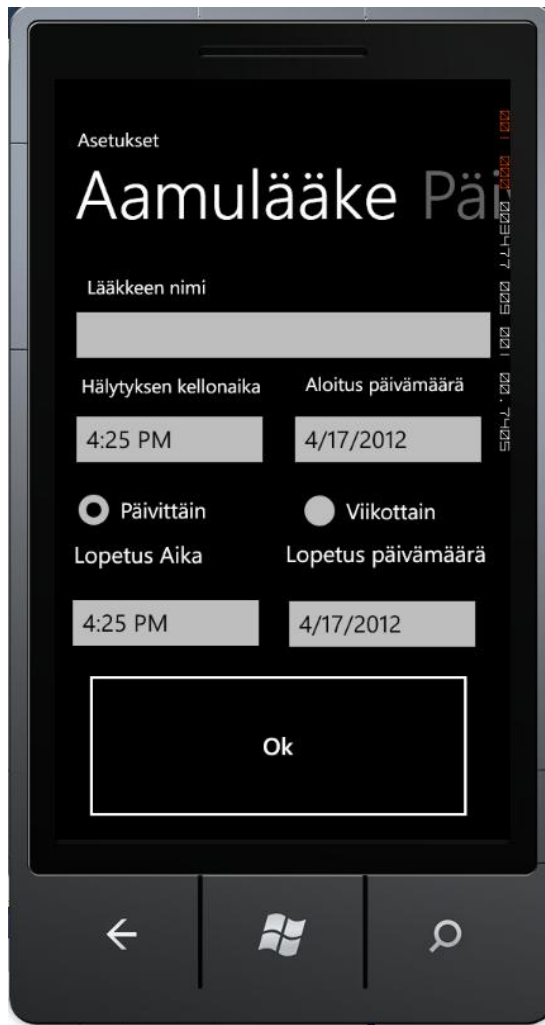
Jotta uusia lääkemuistutuksia voitaisiin ajastaa, piti minun tutkia tarkemmin kuinka Reminder-luokka toimii. Reminderin toiminnallisuuteen löysin hyvän harjoituksen MSDN-sivulta. Harjoituksessa uusia muistutuksia voitiin asettaa käyttäen erillistä asetussivua (Code Samples for Windows Phone 2012, hakupäivä 12.5.2012.) Päätin käyttää tämän tyylistä ratkaisua myös Medicine Time -sovelluksessa.

Muistutukset välilehti on myös suojattu salasanalla, ettei muistutuksia voi muokata kuka tahansa. Muistutukset-sivulla voidaan tarkastella tämänhetkisiä muistutuksia sekä asettaa uusi muistutus painamalla alalaidan "+"-painiketta(kuva16).



KUVA 16: Muistutukset

Sitten käyttäjä pääsee asettamaan uuden muistutuksen tiedot uudella pivot-sivulla(kuva 17), joka sisältää neljä välilehteä. Välilehdet on nimetty seuraavalla tavalla : aamulääke, päivälääke, iltalääke sekä yölläke.



KUVA 17: Uuden muistutuksen lisääminen

Käyttäjä valitsee välilehdistä oikeaa lääkeaikaa vastaavan ja asettaa muistutukselle tiedot. Muistutusta varten täytyy käyttäjän asettaa ensin lääkkeen nimi, alkamisaika, lopetusaika, alkamispäivä sekä lopetuspäivä. Sovellus ilmoittaa, jos tiedot ovat ristiriidassa. Esimerkiksi lopetusaika ei voi olla ennen aloitusaikaa, nimeä ei voi jättää tyhjäksi tai aamulääkettä ei voida ajastaa aamun ulkopuolelle.

Kun uusi muistutus on ajastettu, näkyy se Kuvan 16 muistutussivulla. Jos hälytys halutaan poistaa, onnistuu se painamalla punaista rastia muistutuksen tietojen oikealla puolella.

4.2.5 Käyttöliittymän suunnittelu

Samalla kun kehitin eri osa-alueita, hion ohjelmalle myös käyttöliittymää. Helpottaakseni sovelluksen käytettävyyttä lisäsin siihen alkuvalikon varhaisessa vaiheessa, joka helpottaisi liikkumista sovelluksen eri osa-alueille.

Alkuvalikon ensimmäisessä versiossa oli kolme painiketta, joista yksi vei lokikirjaan, toinen asetuksiin ja kolmas hälyttimeen. Kun sovelluksen kehitys meni pidemmälle, lisäsin pääsyn muistutus-sivulle asetusten kautta. Kyseisen testausversion arkkitehtuuria esittelen tarkemmin luvussa 5.

Kun testausversion eri ominaisuudet olivat testattu, viimeistelin käyttöliittymän poistamalla toimimattomat ominaisuudet, yksinkertaistamalla valikoita sekä lisäämällä aputoiminnon. Esittelen lopullisen version arkkitehtuuria tarkemmin luvussa 5.

4.3 Muutoksia ja ongelmia

Kaikkia määrityksen mukaisia ominaisuuksia ei voitu toteuttaa, johtuen Windows Phone -alustan rajoituksista. Myös muutamia ongelmia ilmeni toteutuksen aikana, ja lopulliseen versioonkin jäi ominaisuuksia, joita ei sovelluksen tekohetkellä voitu ratkaista.

4.3.1 Tekstiviestitoiminnon poisjättäminen

Alkuperäisen suunnitelman mukaan, jos lääke olisi jäänyt ottamatta, lähettäisi puhelin tekstiviestin omaisille, mutta Microsoft on rajoittanut Windows Phone -alustaa siten, että automaattisia tekstiviestejä laite ei voi lähettää ilman että käyttäjä painaa tekstiviestisovelluksen lähetysnappia. Tämä rajoitus on tehty erilaisia roskaposti- ja huijausohjelmia vastaan, etteivät sovellukset esimerkiksi voisi hyväksikäyttää osoitekirjan tietoja ja lähettää salaa mainoksia niiden perusteella.

Päätimme yhdessä Leo Ilkon kanssa, että tämä ominaisuus jätetään pois. Ominaisuus luultavasti hämmentäisi vain käyttäjää, jos puhelin vaatisi lähettämään tekstiviestejä, joissa ilmoitetaan lääkkeen jääneen ottamatta.

4.3.2 SkyDrive-toiminnon tutkiminen

Koska tekstiviestin lähetystoimintoa ei voitu kehittää, aloin selvittämään muita keinoja, miten omaiselle voitaisiin lähettää lokitiedot. Tätä selvittäessäni löysin Microsoftin SkyDrive- palvelun, jossa Windows Live- palvelun käyttäjät voivat lähettää omia tiedostoja Microsoftin pyörittämään pilvipalveluun. Palvelu on maksuton ja sitä voi käyttää luomalla ilmaisen tilin Microsoft Live -palveluun (SkyDrive 2012, hakupäivä 10.4.2012.)

Microsoft julkaisi Live SDK -kehitystyökalupaketin vuoden 2011 lopulla Visual Studioon, jolloin mahdollisuus kehittää SkyDriveä käyttäviä sovelluksia avautui Microsoftin ulkopuolella työskenteleville sovelluskehittäjille.

Tutkittuani asiaa sain kehitettyä sovellukseen toiminnon, joka lähettää SkyDriveen lokitiedot samalla, kun käyttäjä kuittaa hälyttimestä lääkkeenoton. Tämä toiminto toimii ainoastaan, jos käyttäjä käy erikseen kirjautumassa sisään SkyDrive-palveluun käyttäen erillistä kirjautumispainiketta joka kerta kun sovellus käynnistetään (Automatic login with Live Id possible? 2012, hakupäivä 1.5.2012.) Tämä toiminnallisuus on samassa linjassa Microsoftin turvallisuuspolitiikan kanssa, jossa käyttäjän tietämättä ei voida suorittaa automaattisia tietoja lähettäviä ominaisuuksia. SkyDrive-kirjautumispainike nähdään kuvassa 18.



KUVA 18: Sovelluksen testausversion asetusruudussa käyttäjä voi kirjautua SkyDrive-palveluun painamalla painiketta "Sign in"

Kun kirjautumispainiketta on painettu, avautuu kuvassa 14 näkyvä kirjautumisikkuna käyttäjälle, jolla käyttäjä voi kirjautua SkyDrive-palveluun sisään käyttämällä Windows Live -palvelun tunnuksia.



KUVA 19: SkyDrive-palvelun kirjautumisikkuna

Jouduin jättämään tämänkin ominaisuuden lopullisesta versiosta pois, sillä sovellus lisäisi sovelluksen monimutkaisuutta ja aiheuttaisi käyttäjässä hämmennystä, jos hän joutuisi joka kerta ennen lääkkeenottoa painamaan erillistä kirjautumisnappia. Jos Microsoft muuttaa turvallisuuskäytäntöjään ja sallii automaattisen kirjautumisen palveluun, voidaan ominaisuus ottaa käyttöön jatkokehityksessä.

4.3.3 Reminder-toiminnon ongelmat

Muistutusten toteuttaminen käyttäen Reminder-toimintoa ei ollut täysin ongelmatonta. Suurin ongelma liittyy puhelimen ruudun lukitukseen. Jos puhelimen ruutu on lukittuna, ja Reminder-toiminto käynnistyy, ei käyttäjä voi painaa kuvassa 14 näkyvää tekstiä käynnistääkseen Medicine Time

-sovelluksen hälyttimen, vaan jotta hälytin käynnistyisi oikein, täytyy käyttäjän painaa puhelimen Back-painiketta, avata näyttö ja odottaa minuutti, jotta Reminder-toiminto käynnistyy uudestaan. Tämä rajoitus liittyy Microsoftin turvallisuuspolitiikkaan, ettei näyttö lukittuna voi vahingossa käynnistää uusia ohjelmia.

Monet sovelluskehittäjät ovat ilmoittaneet Microsoftille tyytymättömyytensä kyseiseen ominaisuuteen, jonka johdosta saattaa olla, että Microsoft muuttaa käytäntöjään ja toiminto alkaa toimia oikein.

Toinen ongelma Reminder-toiminnon kanssa liittyy siitä avautuvaan hälyttimeen. Alkuperäisessä suunnitelmassa Hälyttimen A-painike kuittasi lääkkeen otetuksi ja B-painike siirsi hälytystä. Nyt jos käyttäjä haluaa siirtää hälytystä, täytyy hänen käyttää Reminder-toiminnon hälytyksen siirtoa, joka nähdään kuvassa 14. Kun käyttäjä painaa tekstiä "OTA LÄÄKE PAINAMALLA TÄSTÄ", kuittaa hän hälytyksen samalla tapahtuneeksi, jolloin hälytinsovelluksessa oleva hälytyksen siirtonappi ei siirrä hälytystä, vaan kirjoittaa lokikirjaan merkinnän "Lääke jäänyt ottamatta".

Ongelmista huolimatta koen onnistuneeni hälyttimen toiminnallisuuden toteuttamisessa hyvin.

5 SOVELLUKSEN TESTAUS JA JULKAISU

Sovelluksen testausta tapahtui koko ajan samanaikaisesti eri osioiden toteuttamisen yhteydessä. Kun uusia ominaisuuksia lisättiin sovellukseen, niiden toimivuutta testattiin ensin emulaattorin avulla, ja sen jälkeen Lumia-älypuhelimissa.

5.1 Sovelluksen testaus emulaattorilla

Visual Studio 2010 -sovelluksen mukana tulevalla emulaattorilla voidaan simuloida Windows Phone -älypuhelinta tietokoneella. Suuri osa Medicine Time -sovelluksen testausta tapahtui käyttäen hyödyksi emulaattoria. Kaikkeen emulaattori ei kuitenkaan pystynyt, kuten säilyttämään lokitietoja uudelleenkäynnistytyn jälkeen, joita varten testausta piti toteuttaa myös Nokia Lumia -älypuhelimilla.

Kun uusia ominaisuuksia lisättiin sovellukseen, testattiin niiden toimivuus ensin emulaattorilla. Jos ohjelmoinnissa oli tapahtunut virheitä, ilmoitti Visual Studio ongelmista jo ennen, kuin emulaattoria voitiin käynnistää. Jos emulaattori käynnistyi, mutta kesken käytön sovellus kaatui ohjelmoinnin virheen vuoksi, simuloi Visual Studio myös emulaattorissa kaatumistilanteen ja ilmoitti, miksi sovellus oli kaatunut. Tällä tavalla testaamalla pystyttiin virheet korjaamaan, jotta loppukäyttäjille ei koituisi ongelmia.

5.2 Sovelluksen testaus testikäyttäjällä

Kun ohjelma oli lähellä valmista, testasin sen käyttöä 27.-30.4.2012 välisenä aikana sovelluksen kohderyhmään kuuluvalla testikäyttäjällä. Testikäyttäjä on 49-vuotias reumaa sairastava nainen, jonka täytyy ottaa sairauteensa lääkitystä aamulla ja illalla. Testilaitteena käytettiin Nokia Lumia 710 -älypuhelinta.

Toimin testausjakson aikana testikäyttäjän "omaisena", jonka johdosta asetin hänelle lääkityksen hälytysajat sekä valvoin lokikirjasta lääkkeenottoja.

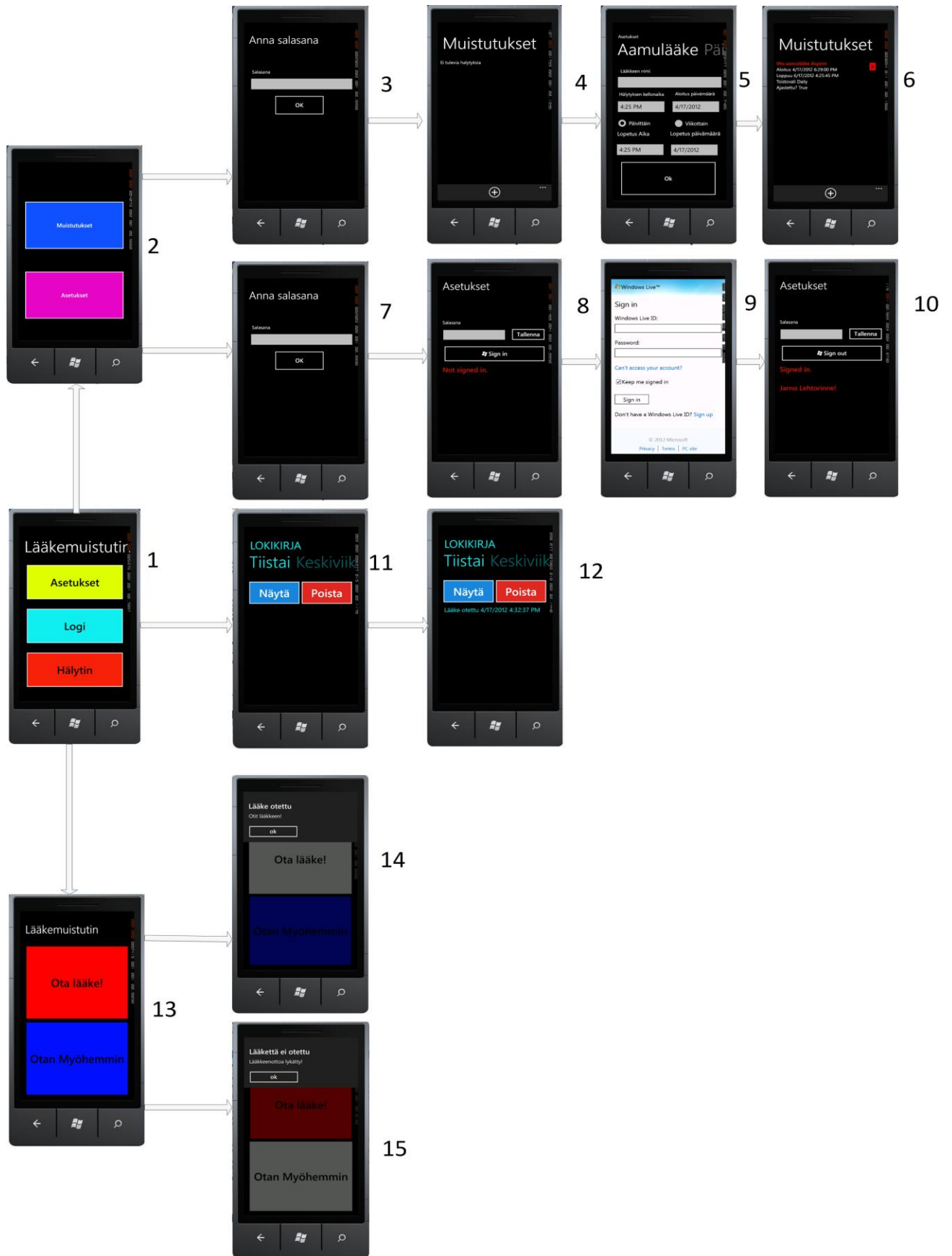
Neuvoin myös häntä sovelluksen käytössä, sillä tässä vaiheessa sovellus ei vielä sisältänyt aputoimintoa. Käyttäjällä oli hieman aikaisempaa kokemusta kosketusnäytöllisistä älypuhelimista, mutta hänellä itsellään ei ole sellaista.

Testausjakson jälkeen haastattelin koekäyttäjää sovelluksesta. Haastattelussa selvisi, että testikäyttäjä oli kokenut sovelluksen helppokäyttöiseksi sekä hyödylliseksi etenkin iltälääkkeen muistutuksen kanssa, joka saattaisi unohtua ilman muistutusta.

Huonona puolena sovelluksessa testikäyttäjä koki sen, ettei muistutuksen kuittaaminen hälytysohjelmalla onnistu, jos näppäimet ovat lukittuna. Arvosteluasteikolla 1-5 testikäyttäjä antoi sovellukselle arvosanan 4.

5.3 Testausversion arkkitehtuuri

Sovelluksen testausversion arkkitehtuuri erosi lopullisesta versiosta jonkin verran. Testikäyttäjällä testattu versio ei sisältänyt SkyDrive-toimintoa, mutta oli muuten kuvassa 20 nähtävän arkkitehtuurin mukainen. Kuvaan on merkattu jokainen sovelluksen ruutu numerolla, ja kuvan alapuolelle on eritelty numeroiden perusteella ruutujen sisältö.



KUVA 20: Testausversion arkkitehtuuri

1. Alkuvalikko, josta voidaan valita, säädetäänkö asetuksia vai mennäänkö lokiin. Kuvassa on testausversion alkuvalikko, mistä pääsee myös hälyttimeen manuaalisesti.
2. Asetusvalikossa valitaan, mennäänkö asettamaan muistutuksia vai muokkaamaan asetuksia.
3. Salasanaruutu kysyy salasanaa, jotta käyttäjä pääsisi asettamaan muistutuksia.
4. Muistutusruutu, kun tulevia muistutuksia ei vielä ole.
5. Lisää uusi muistutus-ruutu. Käyttäjä asettaa lääkkeen tiedot ja hälytysajat tässä ruudussa.
6. Muistutusruutu, jossa näkyy juuri lisätty muistutus.
7. Salasanaruutu kysyy salasanaa, jotta käyttäjä pääsisi siirtymään asetukset välilehdelle.
8. Asetusruutu, jossa voidaan vaihtaa salasana ja kirjautua sisään SkyDrive-palveluun painamalla "Sign in" -painiketta.
9. SkyDrive-palvelun sisäänkirjautumisikkuna. Käyttäjä voi kirjautua sisään käyttämällä Windows Live -palvelun tunnuksia.
10. Asetusruutu kun käyttäjä on kirjautunut sisään. "Sign in" -painike on muuttunut "Sign out" -painikkeeksi, jolla käyttäjä voi kirjautua ulos SkyDrive palvelusta. Ruudun alalaidassa lukee teksti "Hello" ja käyttäjän etunimi sekä sukunimi.

11. Lokikirjan perusnäky. Painamalla näytä-painiketta voidaan tarkastella kyseisen viikonpäivän lokitietoja. Painamalla poista nappia voi käyttäjä tyhjentää lokin.

12. Käyttäjä on painanut lokikirjan näytä-painiketta ja sovellus näyttää tiistain lokitiedot tekstinä.

13. Hälytin. Jos käyttäjä haluaa ottaa lääkkeen, painaa hän punaista "Ota Lääke" -painiketta. Jos käyttäjä jättää lääkkeen ottamatta, painaa hän sinistä "Otan myöhemmin" -painiketta.

14. Käyttäjä on painanut "Ota Lääke" -painiketta ja sovellus kertoo käyttäjälle avautuvassa ikkunassa, että lääke on otettu. Painamalla "Ok"-painiketta hälytin sulkeutuu.

15. Käyttäjä on painanut "Otan myöhemmin" -painiketta, ja sovellus kertoo käyttäjälle avautuvassa ikkunassa, että lääke on jäänyt ottamatta. Painamalla "Ok" -painiketta hälytin sulkeutuu.

5.4 Lopullisen version arkkitehtuuri

Kun testikäyttäjän testijakso oli ohitse, testasin itse vielä SkyDrive-toimintoa, joka jäi automaattisen kirjautumisen puutteesta johtuen pois lopullisesta versiosta.

Lopullisen version arkkitehtuuri erosi testausversiosta myös valikoissa liikkumisen parantamisena, sekä lisäämällä muutaman ominaisuuden lisää. Kuvassa 21 on esitelty lopullisen version arkkitehtuurin erot testausversioon nähden.

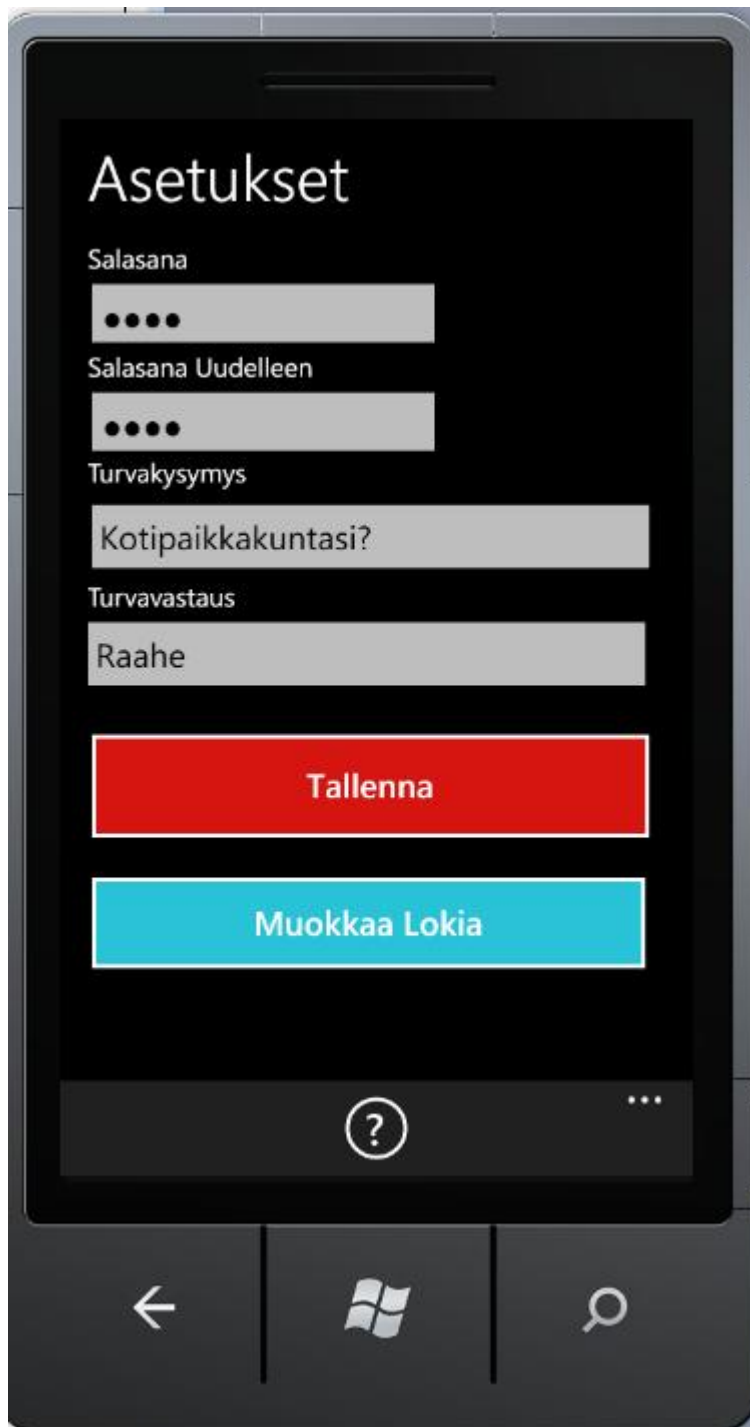
Eroja testausversioon on alkuvalikosta Hälytin-painikkeen poistaminen, ja tilalle on lisätty Muistutukset-painike kuvan kohdassa 1, jolla päästään salasanakyselyyn, jotta omainen voi siirtyä muokkaamaan muistutuksia.

Asetusvalikko on myös poistettu, ja nyt Alkuvalikon Asetus-painike vie suoraan salasanakyselyyn, jotta asetuksiin päästäisiin käsiksi. Käyttäjä voi tarkastella lokia painamalla Loki-painiketta, muttei poistaa sen tietoja kuten testausversiossa. Jos omainen haluaa poistaa lokitietoja, on hänen painettava asetusruudun alalaidassa olevaa Muokkaa lokia -painiketta kuvan kohdassa 3.

Salasanakyselyyn kuvan kohdissa 2 ja 9 on myös lisätty salasanapalautustoiminto turvakysymyksen avulla. Omainen voi asettaa turvakysymyksen ja sille vastauksen asetusruudussa, joka on kuvan kohdassa 3.

Kuvan kohdassa 6 salasanapalautusta varten kysyy sovellus käyttäjältä turvakysymyksen, johon oikein vastaamalla sovellus kertoo oikean salasanan kuvan kohdassa 7.

Kuvan kohdassa 16 on aputoiminto, johon päästään painamalla pientä kysymysmerkkiä ruudun alalaidasta, joka on esitelty kuvassa 22. Aputoiminto helpottaa sovelluksen käyttöönottoa, sillä siinä selvennetään, mitä sovelluksella voidaan tehdä, ja miten.



KUVA 22: Asetusruudun alalaidassa näkyvä kysymysmerkki, joka aukaisee aputoiminnon

5.5 Sovelluksen julkaisu Windows Phone Marketplace -palvelussa

Kun ohjelman testaus oli suoritettu, ja lopullinen versio oli valmistunut, oli aika julkaista se Windows Phone Marketplace -palvelussa.

Sovelluksen julkaisu tapahtui App Hub -sivun kautta samojen askelten mukaisesti kuin olin kuvaillut luvussa 4. Lähetin sovelluksen Microsoftin arvioitavaksi ja asetin sille myyntihinnaksi 0.99€. Sovelluksen ensimmäinen myytävä versio on suomenkielinen ja on myytävänä vain suomenkielisessä Marketplacessa. Käännän sovelluksesta myös englanninkielisen version myöhemmin toukokuussa ja asetan sen myytäväksi ilman aluerajoituksia.

6 JATKOKEHITYSMAHDOLLISUUDET

Sovelluksen jatkokehittämistä varten tutkin kuinka Bluetooth-yhteys voitaisiin luoda ubiPILL-järjestelmän kanssa. Sovellus voisi korvata ubiPILL-järjestelmän mukana kannettavan hälytinosan ja kommunikoisi Bluetoothin kautta päätelaitteen kanssa. Sovellus voisi myös hyödyntää valmiiksi rakennettuja palvelinominaisuuksia.

Ikävä kyllä Microsoft ei ole vielä julkaissut Bluetooth-ominaisuudelle omia kehitystyökaluja, joten Bluetooth-yhteyden lisääminen voi osoittautua hankalaksi. Yksi mahdollisuus ottaa Bluetooth käyttöön on luoda itse oma rekisteritiedosto, jota hyväksikäyttämällä sovellukset voivat lähettää Bluetoothin kautta tietoa. Toinen mahdollisuus on odottaa Microsoftilta virallisia kehitystyökaluja.

Toinen jatkokehitysmahdollisuus kohdistuu lokitietojen lähettämiseen. SkyDrive-ominaisuudella lokitiedot voidaan lähettää omaiselle, mikäli Microsoft muuttaa automaattisen sisäänkirjautumisen käytäntöjään. Myös tekstiviestiominaisuus on mahdollista toteuttaa siten, että sovellus lähettää tietoa erilliselle tietokonepalvelimelle, joka tiedon saatuaan voi lähettää omaiselle tekstiviestin.

7 YHTEENVETO

Tässä opinnäytetyössä tutustuttiin Windows Phone -alustan sovelluskehittämiseen teorian ja käytännön esimerkkien kautta.

Työssä saatiin valmiiksi toimiva lääkemuistutin-sovellus, joka on asetettu myyntiin Marketplaceen, kuten tarkoitus olikin. Työ eteni hyvää vauhtia loppua kohden alku- ja keskivaiheen lievien hidastelujen jälkeen. Eniten vaikeuksia työssä aiheutti erinäiset alustaan liittyvät rajoitukset, joiden tarkoitus on suojella käyttäjiä haittaohjelmilta, mutta samalla ne vaikeuttavat sovelluskehittäjien mahdollisuuksia eri toimintojen toteuttamiseen.

Kokonaisuudessaan Windows Phone on hyvien esimerkkien, selkeiden kirjastojen ja hyvien työkalujen kautta helppo alusta kehittää hienoja sovelluksia älypuhelimille. Opin opinnäytetyön tekemisen aikana paljon uutta itse Windows Phone -alustasta sekä ylipäättänsä sovelluskehittämisestä. Uskon, että tästä projektista saatua kokemusta voin hyödyntää tulevilla sovelluskehitysprojekteilla.

LÄHDELUETTELO

All About WP7 Isolated Storage - Read and Write text files.2011.Hakupäivä 1.2.2012

<http://windowsphonegeek.com/tips/all-about-wp7-isolated-storage-read-and-save-text-files>

Application Certification Requirements for Windows Phone.2012.Hakupäivä 12.5.2012

<http://msdn.microsoft.com/en-us/library/hh184843%28v=VS.92%29.aspx>

Automatic login with Live Id possible?.2012.Hakupäivä 1.5.2012

<http://forums.create.msdn.com/forums/p/102004/605491.aspx>

Background Agents Overview for Windows Phone.2012.Hakupäivä 12.2.2012

[http://msdn.microsoft.com/en-us/library/hh202942\(v=vs.92\).aspx](http://msdn.microsoft.com/en-us/library/hh202942(v=vs.92).aspx)

Code Samples for Windows Phone.2011.Hakupäivä 4.1.2012

<http://msdn.microsoft.com/en-us/library/ff431744%28v=VS.92%29.aspx>

DateTime Structure.2012.Hakupäivä 12.5.2012

<http://msdn.microsoft.com/en-us/library/system.datetime.aspx>

Haikala, Ilkka, Märijärvi Jukka, 2004, Ohjelmistotuotanto, Talentum Media Oy
Johdatus tietojärjestelmiin.2012.Hakupäivä 2.1.2012

http://www.okol.org/verkkokurssit/datanomi/tietojarjestelmien_kaytto_ja_kehittaminen/johdatus_tietojarjestelmiin/kehittamistyon_vaiheet_ja_elikaarimallit/kehittamistyon_vaiheet_ja_elinkaarimallit_asia.htm

Media Element Class.2012.Hakupäivä 12.5.2012

<http://msdn.microsoft.com/en-us/library/system.windows.controls.mediaelement%28v=VS.95%29.aspx>

Microsoft.Devices Namespace.2012.Hakupäivä 12.5.2012

<http://msdn.microsoft.com/en-us/library/microsoft.devices%28v=VS.92%29.aspx>

Microsoft.Phone.Scheduler namespace.2012.Hakupäivä 12.5.2012

<http://msdn.microsoft.com/en-us/library/microsoft.phone.scheduler%28v=vs.92%29.aspx>

Nokia Lumia 710.2012.Hakupäivä 4.5.2012

<http://www.nokia.com/au-en/products/phone/lumia710/specifications/>

Nokia Lumia 800.2012.Hakupäivä 4.5.2012

<http://www.nokia.com/au-en/products/phone/lumia800/specifications/>

Petzold, Charles, 2010, Programming Windows Phone 7, Microsoft Press

Pivot Control Overview for Windows Phone.2012.Hakupäivä 12.5.2012

<http://msdn.microsoft.com/en-us/library/ff941098%28v=vs.92%29.aspx>

Q&A: Microsoft on Windows Phone 7 series.2011. Hakupäivä 2.1.2012.

<http://asia.cnet.com/ganda-microsoft-on-windows-phone-7-series-62061278.htm>

SkyDrive.2012.Hakupäivä 10.4.2012

<http://windows.microsoft.com/fi-FI/skydrive/home>

StreamWriter Class.2012.Hakupäivä 12.5.2012

<http://msdn.microsoft.com/en-us/library/system.io.streamwriter.aspx>

StreamReader Class.2012.Hakupäivä 12.5.2012

<http://msdn.microsoft.com/en-us/library/system.io.streamreader.aspx>

Wikipedia.2012.Hakupäivä 4.5.2012

http://en.wikipedia.org/wiki/Windows_Phone

Wikipedia.2012.Hakupäivä 5.5.2012

http://en.wikipedia.org/wiki/Microsoft_Silverlight

Wikipedia.Hakupäivä 6.5.2012

http://en.wikipedia.org/wiki/Visual_Studio_2010#Visual_Studio_2010

Wikipedia.2012.Hakupäivä 6.5.2012

http://en.wikipedia.org/wiki/Expression_Blend

Windows Phone Marketplace passes 80,000 apps.2012.Hakupäivä 6.4.2012

http://allaboutwindowsphone.com/news/item/14554_Windows_Phone_Marketplace_pass.php

XAML Overview.2012.Hakupäivä 5.5.2012

http://msdn.microsoft.com/en-us/library/ms752059.aspx#what_is_xaml